TESIS

ANALISIS PENERAPAN WORD EMBEDDING TERHADAP BIDIRECTIONAL LSTM DALAM MODEL ANALISIS SENTIMEN ULASAN APLIKASI

Disusun oleh:

Nama	**: Verra Budhi Lestari**
NIM	**: 22.51.1191**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2024

TESIS

ANALISIS PENERAPAN WORD EMBEDDING TERHADAP BIDIRECTIONAL LSTM DALAM MODEL ANALISIS SENTIMEN ULASAN APLIKASI

ANALYSIS OF THE APPLICATION OF WORD EMBEDDING TO BIDIRECTIONAL LSTM ON SENTIMENT ANALYSIS MODEL APPLICATION REVIEWS

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Verra Budhi Lestari**
NIM	**: 22.51.1191**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2024

HALAMAN PENGESAHAN

ANALISIS PENERAPAN WORD EMBEDDING TERHADAP BIDIRECTIONAL LSTM DALAM MODEL ANALISIS SENTIMEN ULASAN APLIKASI

ANALYSIS OF THE APPLICATION OF WORD EMBEDDING TO BIDIRECTIONAL LSTM ON SENTIMENT ANALYSIS MODEL APPLICATION REVIEWS

Dipersiapkan dan Disusun oleh

Verra Budhi Lestari

22.51.1191

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 10 Juli 2024

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 10 Juli 2024
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

ANALISIS PENERAPAN WORD EMBEDDING TERHADAP BIDIRECTIONAL LSTM DALAM MODEL ANALISIS SENTIMEN ULASAN APLIKASI

ANALYSIS OF THE APPLICATION OF WORD EMBEDDING TO BIDIRECTIONAL LSTM ON SENTIMENT ANALYSIS MODEL APPLICATION REVIEWS

Dipersiapkan dan Disusun oleh

Verra Budhi Lestari

22.51.1191

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 10 Juli 2024

Pembimbing Utama

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Anggota Tim Penguji

Dhani Ariatmanto, S.Kom., M.Kom., Ph.D.
NIK. 190302197

Dr. Andi Sunyoto, M.Kom.
NIK. 190302052

Pembimbing Pendamping

Hanafi, S.Kom., M.Eng., Ph.D.
NIK. 190302024

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 10 Juli 2024
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Verra Budhi Lestari
NIM : 22.51.1191
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Analisis Penerapan Word Embedding Terhadap Bidirectional LSTM dalam Model Analisis Sentimen Ulasan Aplikasi

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom.
Dosen Pembimbing Pendamping : Hanafi, S.Kom., M.Eng., Ph.D.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 10 Juli 2024
Yang Menyatakan,

Verra Budhi Lestari

HALAMAN PERSEMBAHAN

Alhamdulillah, puji syukur saya panjatkan kehadirat Allah Subhanahu wa Ta'ala yang telah melimpahkan rahmat dan karunia-Nya. Dengan rasa syukur yang mendalam, penulis mengucapkan terimakasih kepada pihak-pihak yang telah berperan atas terselesaikannya skripsi ini baik secara langsung maipun tidak langsung. Skripsi ini penulis persembahkan kepada :

1. Kedua orang tua, yakni Bapak dan Ibu yang senantiasa selalu mendukung dan mendoakan penulis. Tanpa kalian penulis tidak akan sampai di titik ini sekarang, jasa kalian tidak akan pernah bisa tergantikan dengan apapun di dunia ini.
2. Keluarga, terutama adik penulis yang tidak henti-hentinya untuk selalu memberikan dukungan kepada penulis.
3. Ibu Prof. Ema Utami, S.Si., M.Kom., selaku dosen pembimbing I yang telah banyak membantu, membimbing, dan memberi arahan kepada penulis dalam menyelesaikan penelitian thesis ini.
4. Bapak Hanafi, S.Kom., M.Eng., Ph.D. selaku dosen pembimbing II penulis yang selalu membimbing dan memberi arahan selama penyusunan thesis.
5. Rekan dan sahabat penulis, Tiara, Ayu, Sofyan, Nium, Anti, Fidya, Rusdi, Anip, dan Handa yang tidak henti-hentinya memberikan motivasi untuk saya serta menemani saya dalam suka dan duka, serta Chiki dan Gems yang selalu menghibur.

6. Teman-teman saya, khususnya MTI angkatan 2022 yang telah berjuang bersama selama masa perkuliahan.
7. Dan tidak lupa untuk diri saya sendiri, terimakasih untuk selalu memiliki semangat yang tinggi, selalu berusaha dan tidak menyerah.

HALAMAN MOTTO

"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya. Dia mendapat (pahala) dari (kebajikan) yang dikerjakannya dan dia mendapat (siksa) dari (kejahatan) yang diperbuatnya."

(QS. Al-Baqarah : 286)

"Selalu ada harga dalam sebuah proses. Nikmati saja lelah-lelah itu. Lebarkan lagi rasa sabar itu. Semua yang kau investasikan untuk menjadikan dirimu serupa yang kau impikan, mungkin tidak akan selalu berjalan lancar. Tapi, gelombang-gelombang itu yang nanti bisa kau ceritakan."

- Boy Chandra

"Tidak ada kesuksesan tanpa kerja keras. Tidak ada keberhasilan tanpa kebersamaan. Dan tidak ada kemudahan tanpa doa."

- Ridwan Kamil

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan nikmat dan rahmat-Nya sehingga penulis dapat menyelesaikan penulisan thesis ini dengan judul **"Analisis Penerapan Word Embedding Terhadap Bidirectional LSTM dalam Model Analisis Sentimen Ulasan Aplikasi"**.

Adapun pengajuan thesis ini disusun untuk memenuhi salah satu syarat untuk memperoleh gelar Magister dalam Program Strata-2 Informatika di Universitas Amikom Yogyakarta. Selama mengikuti perkuliahan hingga penyusunan thesis ini, banyak hambatan dan tantangan serta kesulitan yang penulis alami, namun berbagai pihak telah membantu penulis. Untuk itu penulis ingin menyampaikan terimakasih kepada semua pihak yang telah terlibat serta membantu penulis, khususnya kepada :

1. Allah Subhanallahu wa ta'ala yang telah memberikan berkah, rahmat dan hidayah-Nya sehingga thesis ini dapat terselesaikan.
2. Kedua Orang Tua penulis, Bapak Warsisno dan Ibu Desi Hernawati, serta saudara saya tercinta Hannita Dwi Lestari, atas segala do'a dan dukungan yang tidak henti-hentinya diberikan selama ini.
3. Bapak Prof. Dr. M. Suyanto, MM., selaku Rektor Universitas Amikom Yogyakarta.
4. Ibu Prof. Ema Utami, S.Si., M.Kom. dan Bapak Hanafi, S.Kom., M.Eng., Ph.D. selaku dosen pembimbing yang telah banyak membantu

penulis serta selalu memberikan bimbingan dan arahan dengan sangat baik.

5. Bapak Dhani Ariatmanto, S.Kom., M.Kom., Ph.D. dan Bapak Dr. Andi Sunyoto, M.Kom. selaku dosen penguji yang telah memberikan saran serta arahan.
6. Dosen dan Staff Pascasarjana Universitas Amikom Yogyakarta yang telah berperan dalam membimbing dan mendukung proses perkuliahan penulis.
7. Serta seluruh pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu serta mendukung penulis baik dalam penyusunan thesis ini maupun selama perkuliahan.

Penulis menyadari bahwa dalam pembuatan thesis ini masih terdapat kekurangan dan kelemahan. Oleh karena itu penulis berharap kepada semua pihak agar dapat menyampaikan kritik dan saran yang membangun untuk menambah kesempurnaan penulisan thesis ini. Namun penulis tetap berharap thesis ini dapat memberikan manfaat bagi semua pihak dan para pembaca.

Yogyakarta, 10 Juli 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Analisis sentimen aplikasi bertujuan untuk mengetahui polaritas sentimen ulasan aplikasi berdasarkan umpan balik dari pengguna dengan data dalam jumlah besar. Berbagai metode *machine learning*, *neural network*, hingga *deep learning* digunakan dalam membangun model analisis sentimen, namun beberapa masih memiliki kekurangan terkait lambatnya model dalam mencapai konvergensi maupun permasalahan mengenai *out of vocabulary* (OOV). Sehingga diperlukan penerapan *word embedding* sebagai upaya dalam meningkatkan keakuratan klasifikasi. Selain itu penerapan *word embedding* dinilai dapat menghasilkan representasi kata yang lebih banyak dan variatif. *Word embedding* yang diterapkan dalam penelitian ini adalah Word2Vec, FastText, dan GloVe, hal ini untuk mengetahui model *word embedding* serta parameter yang mampu menghasilkan performa yang unggul saat dikombinasikan dengan Bi-LSTM dalam melakukan analisis sentimen terhadap ulasan pengguna aplikasi Gojek.

Pada kondisi *imbalanced* dataset maupun setelah *oversampling* performa terbaik dihasilkan saat penerapan Word2Vec dan FastText dengan parameter *window* bernilai 5, dan *embedding size* untuk Word2Vec, FastText, dan GloVe bernilai 300. Perolehan *f1-score* pada *imbalanced* dataset yang dihasilkan Word2Vec, FastText, dan GloVe sebesar 86,03%, 86,29%, 84,54%, sementara akurasi pada data setelah *oversampling* sebesar 87,51%, 87,85%, 85,98%.

Model FastText menghasilkan performa *f1-score* dan akurasi yang unggul baik pada parameter tertentu maupun secara rata-rata karena mampu menangani kata-kata yang memiliki morfologi kompleks maupun *out of vocabulary* dengan cara merepresentasikan kata tidak hanya secara penuh namun juga memperhatikan sub-kata. Selain *window* dan *embedding size*, perolehan performa tersebut juga dipengaruhi oleh penerapan beberapa parameter lain, seperti arsitektur CBOW, metode evaluasi *Hirarchical SoftMax*, dan *minimum count*=10.

Kata kunci: Analisis Sentimen, Bi-LSTM, Word2Vec, FastText, GloVe

ABSTRACT

Application sentiment analysis aims to determine the sentiment polarity of application reviews based on feedback from users with large amounts of data. Various machine learning, neural network, and deep learning methods are used to build sentiment analysis models, but some still have shortcomings related to the slowness of the model in achieving convergence and problems regarding out of vocabulary (OOV). So, it is necessary to apply word embedding as an effort to increase classification accuracy. Apart from that, the application of word embedding can produce more and more varied word representations. The word embeddings applied in this research are Word2Vec, FastText, and GloVe, this is to determine the word embedding model and parameters that can produce superior performance when combined with Bi-LSTM in conducting sentiment analysis on user reviews of the Gojek application.

In the imbalanced dataset and after oversampling, the best performance is generated when applying Word2Vec and FastText with a window parameter of 5, and embedding size for Word2Vec, FastText, and GloVe is 300. The f1-score on the imbalanced dataset generated by Word2Vec, FastText, and GloVe is 86.03%, 86.29%, 84.54%, while the accuracy on the data after oversampling is 87.51%, 87.85%, 85.98%.

The FastText model produces the best f1-score and accuracy performance both on certain parameters and on average because it is able to handle words that have complex morphology and out of vocabulary by representing words not only in full but also paying attention to sub-words. Besides window and embedding size, the performance gain is also influenced by the application of several other parameters, such as CBOW architecture, Hierarchical SoftMax evaluation method, and minimum count=10.

Keyword: Sentiment Analysis, Bi-LSTM, Word2Vec, FastText, GloVe

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Analisis sentimen telah menjadi topik penelitian hangat dalam pemrosesan bahasa alami dan bidang penambangan data dalam dekade terakhir (Basiri, Nemati, Abdar, Cambria, & Acharrya, 2021) (Sangeetha & Kumaran, 2022). Analisis sentimen adalah salah satu tugas utama *Natural Language Processing* (NLP), di mana sikap, pemikiran, pendapat, atau penilaian terhadap subjek tertentu telah diekstraksi, pengakuan sentimen dapat bermanfaat bagi pembuat keputusan individu, organisasi bisnis, dan pemerintah (Onan, 2020).

Analisis sentimen mengklasifikasikan dokumen teks kedalam kelompok-kelompok sentimen, seperti positif, netral, dan negatif, pengelompokan ini dapat dilakukan pada persepsi masyarakat dan dimanfaatkan untuk mengetahui opini masyarakat terhadap suatu topik (Nurdeni, Budi, & Santoso, 2021). Analisis sentimen juga mampu melakukan klasifikasi terhadap emosi seseorang melalui teks (Kharde & Sonawane, 2016).

Beberapa topik yang menjadi pembahasan pada analisis sentimen, antara lain mengenai suatu fenomena, seperti Covid19 maupun Vaksin Covid19 (Xiaoyan, Raga, & Xuemei, 2022), *cyberbullying* (Alsharef, et al., 2022), hingga deteksi emosi melalui ungkapan pada media sosial (Riza & Charibaldi, 2021). Selain itu analisis sentimen juga membahas mengenai pendapat masyarakat yang dituangkan dalam ulasan berbentuk teks, seperti ulasan produk *marketplace* (Alharbi,

Alghamdi, Alkhammash, & Al Amri, 2021) (Yang, Li, Wang, & Siherratt, 2020) (Zhao, Liu, Yao, & Yang, 2021) (Iqbal, et al., 2022), ulasan aplikasi (Rahman, Utami, & Sudarmawan, 2021) (Alghifari, Edi, & Firmansyah, 2022) (Aljedaani, Rustam, Ludi, Ouni, & Mkaouer, 2021), ulasan obat (Colón-Ruiz & Segura-Bedmar, 2020), hingga ulasan film (Af'idah, Handayani, & Pratiwi, 2021) (Subba & Kumari, 2022) (Cahyo, et al., 2023).

Seperti halnya suatu produk, aplikasi juga harus mampu memenuhi hal-hal apa saja yang dibutuhkan oleh pengguna, hal ini karena aplikasi membutuhkan pengembangan agar mampu bersaing dengan aplikasi sejenis lainnya. Dengan adanya fitur ulasan yang tersedia pada platform penyedia aplikasi seperti Google PlayStore, App Store, Amazon dan lainnya, pengguna dapat menyampaikan pendapat mereka mengenai kualitas, fitur, layanan, hingga tampilan aplikasi tersebut melalui ulasan. Data ulasan pengguna inilah yang akan digunakan sebagai salah satu dasar penentu keputusan dalam mengembangkan aplikasi tersebut melalui analisis sentimen (Gondhi, et al., 2022).

Analisis sentimen aplikasi maupun produk bertujuan untuk menentukan polaritas sentimen pada ulasan aplikasi dengan kondisi data dengan jumlah yang besar. Hasil analisis sentimen dimanfaatkan untuk mengetahui bagaimana opini pengguna mengenai aplikasi tersebut, bagaimana aplikasi tersebut diterima oleh masyarakat, serta hal apa yang perlu dikembangkan oleh developer pada aplikasi tersebut, tentunya berdasarkan *feedback* dari pengguna (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) (Sangeetha & Kumaran, 2022).

Beberapa penelitian terkait analisis sentimen ulasan aplikasi telah banyak dilakukan dengan menerapkan berbagai metode dengan hasil sentimen yang cukup beragam. Salah satunya pada penelitian analisis sentimen aplikasi Grab, (Alghifari, Edi, & Firmansyah, 2022) menerapkan metode Bi-LSTM serta beberapa metode *machine learning* (Multinomial Naive Bayes, Logistic Regression, dan Linear Support Vector Machine) sebagai pembandingnya. BI-LSTM dikombinasikan dengan ekstraksi fitur dan pembobotan *Bag of Word* (BoW) dan mampu unggul dibandingkan metode lainnya dengan akurasi sebesar 91% dan training loss 28%. Peneliti mengungkapkan Bi-LSTM membutuhkan dataset yang besar untuk menghindari permasalahan *overfitting*, sehingga diperlukan penerapan kombinasi word embedding untuk memperoleh hasil representasi kata yang lebih banyak dan bervariasi.

Penelitian (Aljedaani, Rustam, Ludi, Ouni, & Mkaouer, 2021) menerapkan model *machine learning*: *Logistic Regression*, *Support Vector Machine*, *Extra Tree*, *Gaussian Naïve Bayes*, *Gradient Boosting*, dan *Ada Boost* terhadap data ulasan berbahasa Inggris mengenai aksesbilitas aplikasi *opensource* Android untuk mendeteksi polaritas sentimen ulasan suatu aplikasi.

Penelitian lainnya dilakukan oleh (Zhao, Liu, Yao, & Yang, 2021) mengusulkan algoritma *machine learning* berbasis *neural network* yang dioptimalkan yaitu *Local Search Improvised Bat Algorithm based Elman Neural Network* (LSIBA-ENN) untuk melakukan analisis sentimen mengenai ulasan aplikasi yang diperoleh dari situs Amazon.com. Algoritma yang diusulkan

dikombinasikan dengan beberapa word embedding seperti Word2Vec, TF, TF-IDF, TF-DFS, dan *word embedding* yang dikembangkannya yaitu LTF-MICF.

Dalam penelitiannya, (Gondhi, et al., 2022) menggabungkan model *deep learning* LSTM dengan ekstraksi fitur *word embedding* Word2Vec untuk melakukan analisis sentimen mengenai ulasan *e-commerce* Amazon. Penerapan LSTM dengan *word embedding* Word2Vec dalam dataset yang besar dengan kondisi yang tidak seimbang memperoleh hasil akurasi sebesar 89%, presisi 97%, recall 90%, dan f1-score 93%. Hasil ini menunjukan bahwa LSTM dengan ekstraksi fitur mampu unggul dibandingkan Naïve Bayes, *Support Classification*, hingga CNN.

Penelitian (Alsharef, et al., 2022) mengungkapkan bahwa penambahan *word embedding* BERT dan GloVe terhadap model LSTM mampu meningkatkan keakuratan mengenai klasifikasi *toxicity* (toxic dan nontoxic) pada forum obrolan online. Berdasarkan penerapan model LSTM dengan *word embedding* BERT pada penelitian diperoleh hasil akurasi sebesar 94% dan F1-score sebesar 0,89. Sehingga dapat disimpulkan bahwa penambahan *word embedding* mampu meningkatkan keakuratan klasifikasi, kemudian kombinasi LSTM-BERT memiliki kinerja yang lebih baik dibandingkan LSTM tanpa *word embedding* dan LSTM-GloVe.

Penelitian (Riza & Charibaldi, 2021) menerapkan metode LSTM dengan *word embedding* FastText, Word2Vec, dan GloVe dalam melakukan deteksi emosi pada teks yang berasal dari twitter. Penerapan Word2Vec dan FastText mendapatkan akurasi terbaik dalam penelitian ini, yakni sebesar 73,15%.

Sedangkan penerapan GloVe menghasilkan akurasi sebesar 60,10%. FastText dinilai unggul karena mampu menangani masalah *out of vocabulary* (OOV).

Beberapa penelitian lainnya menerapkan model Bi-LSTM dalam melakukan analisis sentimen. Seperti pada penelitian (Subba & Kumari, 2022) Bi-LSTM mampu memperoleh akurasi 92% pada dataset IMDB Reviews saat dikombinasikan dengan BERT. Selain itu Bi-LSTM juga mampu unggul dibandingkan RNN, CNN, Naïve Bayes, dan LSTM dengan F1-Score sebesar 92,18 dalam penelitian (Xu, Meng, Qiu, Yu, & Wu, 2019) dan BiLSTM juga mampu unggul saat diusulkan dengan berbasis *Cascaded Attention Fusion* Model dalam penelitian (Sangeetha & Kumaran, 2022) dengan F1-score sebesar 95,90.

Di antara berbagai arsitektur *Neural Network* yang diterapkan untuk analisis sentimen, model *Long Short-term Memory* (LSTM) dan variannya seperti *Gated Recurrent Unit* (GRU) semakin menarik perhatian (Basiri, Nemati, Abdar, Cambria, & Acharrya, 2021). Namun model LSTM masih memiliki kekurangan terkait lambatnya model dalam mencapai konvergensi. Serta hanya dapat menangkap informasi dari satu arah sehingga diperlukan penerapan metode *deep learning* lainnya seperti Bi-LSTM untuk meningkatkan performa (Alsharef, et al., 2022) (Gondhi, et al., 2022) (Riza & Charibaldi, 2021).

BiLSTM telah diterapkan dalam analisis sentimen untuk berbagai Bahasa, seperti Bahasa Arab (Abdelhady, Soliman, & Farghally, 2024) (Kaibi, Nfaoui, & Satori, 2019), Bahasa China (Xu, Meng, Qiu, Yu, & Wu, 2019), Bahasa Inggris (Colón-Ruiz & Segura-Bedmar, 2020) (Sangeetha & Kumaran, 2022) (Xiaoyan, Raga, & Xuemei, 2022), hingga Bahasa Indonesia. Meskipun telah diterapkan juga

untuk analisis sentimen Bahasa Indonesia, namun data yang digunakan dalam penelitian dapat dikatakan cukup sedikit, yaitu berjumlah dibawah 10.000 (Alghifari, Edi, & Firmansyah, 2022). Penelitian yang akan dilakukan akan menerapkan model Bi-LSTM untuk mengetahui performa model tersebut dalam analisis sentimen Bahasa Indonesia dengan peningkatan jumlah data.

Selain itu untuk dapat menghasilkan representasi kata yang lebih banyak dan variatif dapat dilakukan dengan cara mempertimbangakan kombinasi pada *word embedding*nya (Alghifari, Edi, & Firmansyah, 2022) (Zhao, Liu, Yao, & Yang, 2021). Beberapa penelitian mengenai sentimen analisis ulasan aplikasi mengkombinasikan Bi-LSTM dengan salah satu fitur ekstraksi atau *word embedding*, seperti penelitian (Xu, Meng, Qiu, Yu, & Wu, 2019) yang menerapkan Bi-LSTM dengan kombinasi Word2Vec, peneltian (Xiaoyan, Raga, & Xuemei, 2022) dengan kombinasi Bi-LSTM dan GloVe, penelitian (Abdelhady, Soliman, & Farghally, 2024) dengan Bi-LSTM dan FastText, penelitian (Alghifari, Edi, & Firmansyah, 2022) dengan kombinasi Bi-LSTM dan BoW, serta beberapa penelitian lainnya masih menggunakan *machine learning* (Putri & Khasirudin, 2022) (Rahman, Utami, & Sudarmawan, 2021).

Word2Vec, FastText, GloVe, (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) (Nurdin, Aji, Bustamin, & Abidin, 2020) (Riza & Charibaldi, 2021) merupakan jenis *word embedding* yang banyak dikombinasikan dengan berbagai algoritma *machine learning* hingga *deep learning*, selain *word embedding* tersebut (Alsharef, et al., 2022) (Colón-Ruiz & Segura-Bedmar, 2020) (Ranjan & Daniel, 2022) juga menerapkan BERT dalam penelitianya.

Selanjutnnya penelitian (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) menyatakan *word embedding* memiliki performa yang berbeda saat dikombinasikan dengan masing-masing algoritma, sehingga ekstraksi fitur dan *machine learning* dalam kasus analisis sentimen tidak dapat dinilai secara individual. Sehingga penerapan algoritma harus disesuaikan berdasarkan kebutuhan, yakni berdasarkan dataset dan permasalahan yang ingin diselesaikan. (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) (Nurdin, Aji, Bustamin, & Abidin, 2020).

Terdapat beberapa kelemahan pada penelitian sebelumnya, seperti belum diterapkannya model LSTM dua arah (Aljedaani, Rustam, Ludi, Ouni, & Mkaouer, 2021) (Zhao, Liu, Yao, & Yang, 2021) maupun masih menerapkan model *machine learning* (Alsharef, et al., 2022) (Gondhi, et al., 2022) (Riza & Charibaldi, 2021) sehingga memiliki kekurangan terkait lambatnya model dalam mencapai konvergensi. serta hanya dapat menapkap informasi dari satu arah. Selain itu, pada penelitian lainnya belum dilakukan penerapan variasi *word embedding* terhadap model BiLSTM (Alghifari, Edi, & Firmansyah, 2022). Serta (Af'idah, Handayani, & Pratiwi, 2021) menyarankan untuk menerapkan model *word embedding* lain seperti FastText dan GloVe disarankan untuk diterapkan sebagai pembanding dari Word2Vec.

Berdasarkan beberapa penelitian diatas, penelitian ini akan menerapkan dan membandingkan beberapa metode *word embedding*, yaitu Word2Vec, FastText, dan GloVe, hal ini untuk mengetahui algoritma pembobotan fitur yang mampu menghasilkan performa yang unggul dari Bi-LSTM dalam melakukan analisis

sentimen terhadap ulasan pengguna aplikasi berbahasa Indonesia pada Google Play Store. Evaluasi performa dari algoritma akan dilakukan dengan akurasi, presisi, *recall*, dan *F1-score*. Selain itu penelitian ini juga bertujuan untuk mengetahui pada kondisi (parameter) seperti apa model mampu menghasilkan kinerja yang unggul pada masing-masing penerapan *word embedding*.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah disajikan dapat dirumuskan beberapa rumusan masalah sebagai berikut.

a. Bagaimana hasil performa (akurasi, presisi, *recall, f1-score*) yang diperoleh dari penerapan masing-masing *word embedding* Word2Vec, FastText, dan GloVe saat dikombinasikan dengan Bi-LSTM?
b. Pada kondisi (parameter) seperti apa masing-masing *word embedding* (Word2Vec, FastText, dan GloVe) memiliki hasil performa yang unggul?

1.3. Batasan Masalah

Bagian ini memuat penjelasan tentang:

a. Dataset yang digunakan merupakan data ulasan berbahasa Indonesia yang diperoleh dari ulasan pengguna Google PlayStore terhadap aplikasi Gojek.
b. Dataset yang digunakan merupakan dataset publik yang diperoleh pada situs Kaggle.com.
c. *Pre-processing* yang digunakan adalah *clean text, tokenization, stemming, dan stopword removal*.

d. Analisis sentimen diklasifikasikan dalam 2 kelas (positif dan negatif).
e. Dalam pemodelan dilakukan pembagian dataset dengan *stratified sampling*, sebesar 80% sebagai *training data* dan 20% sebagai *testing data*.
f. Penanganan kondisi dataset yang tidak seimbang dilakukan dengan menerapkan metode *oversampling* SMOTE (*Synthetic Minority Oversampling Technique*).
g. *Word embedding* yang digunakan adalah Word2Vec, FastText, dan GloVe.
h. Penerapan parameter *Word Embedding* dilakukan untuk mengetahui kondisi model dengan performa yang unggul, hal ini tidak berfokus pada pada analisa pengaruh parameter.
i. Metode yang dilakukan untuk melakukan klasifikasi analisis sentimen adalah *Bidirectional* LSTM.
j. Metode evaluasi yang dilakukan dengan *confusion matrix* (akurasi, presisi, *recall, F1-measure*).
k. Penelitian ini menggunakan Google Colaboratory sebagai *platform* penelitian.
l. Hasil yang diukur dari penerapan algoritma *word embedding* dalam melakukan analisis sentimen adalah akurasi, presisi, *recall*, dan *F1-Score*.

1.4. Tujuan Penelitian

Bagian ini memuat penjelasan secara spesifik:

a. Mengetahui hasil performa (akurasi, presisi, recall, f1-score) penerapan word embedding Word2Vec, FastText, dan GloVe dengan algoritma *Bidirectional* LSTM dalam melakukan analisis sentimen ulasan pengguna Google PlayStore terhadap aplikasi Gojek.

b. Mengetahui kondisi (parameter) yang memiliki performa terbaik pada penerapan masing-masing *word embedding* (Word2Vec, FastText, dan GloVe) dalam melakukan analisis sentimen.

1.5. Manfaat Penelitian

Bagian ini memuat penjelasan tentang:

a. Manfaat Teoritis, mengembangkan penelitian mengenai penerapan *word embedding* Word2Vec, FastText, dan GloVe pada metode Bi-LSTM dalam melakukan analisis sentimen ulasan pengguna Google PlayStore terhadap aplikasi Gojek.

b. Manfaat Praktis, berkontribusi secara ilmiah dalam mengembangkan algoritma analisis sentimen pada ulasan pengguna Google PlayStore terhadap aplikasi Gojek menggunakan *word embedding* yang dikombinasikan dengan Bi-LSTM secara lebih efektif.

c. Mengetahui pola ulasan pengguna Google PlayStore terhadap aplikasi Gojek.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Terdapat penelitian terdahulu yang memiliki relevansi terhadap penelitian yang akan dilakukan serta sebagai studi literatur adalah sebagai berikut.

Penelitian yang dilakukan oleh (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) mengusulkan RNN yang digabungkan dengan varian RNN lainnya, seperti *Long Short-Term Memory Networks* (LRNN), *Group Long Short-Term Memory Networks* (GLRNN), *Gated Recurrent Unit* (GRNN), dan *Update Recurrent Unit* (UGRNN) dalam memprediksi pendapat pelanggan berdasarkan ulasan ponsel pada situs Amazon.com yang berjumlah 400.000 ulasan. Kemudian dilakukan perbandingan pada tiap model mengenai penerapan ekstraksi fitur GloVe, Word2Vec, dan FastText untuk analisis sentimen pada data seimbang dan tidak seimbang. Hasil eksperimen menujukan algoritma GLRNN dengan ektraksi fitur FastText unggul pada kondisi data tidak seimbang dengan akurasi sebesar 93,75%, dan pada kondisi data seimbang algoritma LRNN unggul dengan akurasi 88,39%, (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) menilai word embedding memiliki peran yang penting dalam klasifikasi teks.

Kombinasi antara GloVe sebagai *word embedding*, CNN sebagai representasi karakter, dan BiLSTM untuk membangun hubungan temporal yang dilakukan oleh (Xiaoyan, Raga, & Xuemei, 2022) terhadap data yang terdiri dari *complete-text-dataset*, *long-text-dataset*, dan *short-text-dataset*. Dataset yang

digunakan dalam penelitian ini merupakan data komentar pada media sosial twitter mengenai COVID-19 yang berjumlah sebanyak 81.696 komentar, yang terdiri dari 35.093 komentar positif, 31.060 komentar negatif, dan 15.543 komentar netral. Hasil penelitian menunjukan bahwa dengan diterapkannya GloVe sebagai word embedding mampu menghasilkan performa yang paling unggul pada ketiga kondisi dataset dibandingkan dengan model TextCNN dan model CNN-BiLSTM tanpa GloVe dengan akurasi sebesar 0,9565 pada *complete-text-dataset*, 0,9509 pada *long-text-dataset*, dan 0,9560 pada *short-text-dataset*.

Metode Bi-LSTM juga diterapkan pada analisis sentimen aplikasi Grab berdasarkan ulasan pengguna pada Playstore (Alghifari, Edi, & Firmansyah, 2022). Pada penelitian ini Bi-LSTM dikombinasikan dengan ekstraksi fitur dan pembootan *Bag of Words* (BoW), selain itu disajikan juga beberapa algoritma *machine learning*, seperti Multinomial *Naive Bayes*, *Logistic Regression*, dan *Linear Support Vector Machine* sebagai pembanding. Proses pelabelan yang diterapkan terhadap 5.000 data ulasan ini dilakukan berdasarkan rating yang diberikan pengguna, rating 1 dan 2 untuk negatif, rating 3 untuk netral, rating 4 dan 5 untuk positif. Berdasarkan hasil eksperimen, Bi-LSTM dengan BoW mampu unggul dengan akurasi sebesar 91% dan *training loss* 28%, meskipun demikian Bi-LSTM dinilai membutuhkan dataset yang besar untuk menghindari masalah *overfitting*, sehingga untuk memperoleh hasil representasi kata yang lebih banyak dan bervariasi disarankan untuk mempertimbangkan penerapan kombinasi word embedding.

Penelitian (Nurdin, Aji, Bustamin, & Abidin, 2020) membandingkan kinerja *word embedding* Word2Vec, GloVe, dan FastText yang dikombinasikan bersamaan dengan metode CNN untuk melakukan analisis sentimen pada dataset 20 *Newsgroup* dan *Reuters Newswrite*. Dataset 20 *Newsgroup* terdiri atas 11.314 data latih dan 7.532 data uji, sedangkan dataset *Reuters Newswrite* terdiri atas 8.982 data latih dan 2.246 data uji. Hasil perbandingan menunjukkan ketiga *word embedding* memiliki performa yang kompetitif dengan selisih yang tidak terlalu signifikan antara satu dengan lainnya Pada dataset 20 *Newsgroup* diperoleh F-*Measure* sebesar 0,979 utnuk FastText, 0,925 untuk Word2Vec, dan 0,958 untuk GloVe. Hal ini menunjukkan penerapan setiap *word embedding* bergantung pada kondisi dataset dan domain.

Penerapan LSTM sebagai metode untuk melakukan deteksi emosi berdasarkan dokumen teks yang dilakukan oleh (Riza & Charibaldi, 2021) diterapkan dengan kombinasi *word embedding* FastText, Word2Vec, dan GloVe. Deteksi ini dilakukan terhadap 1304 data teks yang berasal dari twitter, dataset terdiri atas 250 happy data, 250 sad data, 200 scared data, 200 disgust data, 204 angry data, dan 201 shocking data. Hasil ekperimen yang diperoleh LSTM dengan kombinasi Word2Vec dan FastText menghasilkan akurasi sebesar 73,15%, sedangkan LSTM dengan kombinasi GloVe hanya mengahsilkan akurasi sebesar 60,10%, FastText dinilai sedikit lebih unggul karena mampu menangani permasalahan *out of vocabulary*. Meskipun lebih unggul hasil akurasi ini dinilai belum cukup baik, sehingga disarankan untuk menambah jumlah dataset yang

digunakan untuk melakukan deteksi dan menerapkan beberapa metode *Deep Learning* lainnya seperti CNN dan Bi-LSTM.

Dalam penelitiannya, (Gondhi, et al., 2022) menggabungkan model *deep learning* LSTM dengan ekstraksi fitur *word embedding* Word2Vec untuk melakukan analisis sentimen mengenai ulasan e-commerce menggunakan dataset berjumlah 938.261 dari Amazon. Penerapan LSTM dengan word embedding Word2Vec dalam dataset yang besar dengan kondisi yang tidak seimbang memperoleh hasil akurasi sebesar 89%, presisi 97%, recall 90%, dan f1-score 93%. Hasil ini menunjukan bahwa LSTM dengan ekstraksi fitur mampu unggul dibandingkan *Naïve Bayes*, *Support Classification*, hingga CNN. (Gondhi, et al., 2022) mengusulkan penerapan LSTM dua arah untuk klasifikasi sentimen pada penelitan yang akan datang.

Word embedding GloVe dan BERT digunakan pada penelitian ini sebagai upaya untuk meningkatkan akurasi mengenai klasifikasi *toxicity* yang dilakukan terhadap aktivitas pada forum obrolan online (Alsharef, et al., 2022). Klasifikasi ini menerapkam model *Neural Network* LSTM dengan 2 kelas klasifikasi (*toxic* dan *non-toxic*). Terdapat 2 skenario klasifikasi yaitu LSTM+BERT dan LSTM+GloVe, masing-masing skenario memperoleh akurasi 94% dan 93% dan F1-score 0,894 dan 0,841. Peneliti mengklaim *word embedding* mampu meningkatkan performa LSTM karena LSTM dengan kombinasi BERT mampu unggul dibandingkan LSTM tanpa *word embedding*, serta disarankan untuk menerapkan LSTM dua arah pada penelitian selanjutnya.

Penelitian (Af'idah, Handayani, & Pratiwi, 2021) menerapkan model CNN dan *word embedding* Word2Vec dalam melakukan analisis sentimen ulasan film untuk mengetahui pengaruh parameter Word2Vec terhadap performa *deep learning*. Parameter yang digunakan adalah Arsitektur: *Continuous Bag of Words* dan *Skip-Gram*, metode evaluasi: *hierarchical softmax* dan *negative sampling*, dan nilai dimensi: 100, 200, dan 300. Hasil penelitian yang dilakukan terhadap 10.000 data ulasan menunjukkan rata-rata performa terbaik dihasilkan oleh penerapan parameter CBOW, *hierarchical softmax*, dan dimanesi bernilai 100. Meskipun demikian, performa parameter lainnya tidak memiliki perbedaan yang jauh sehingga masih cukup kompetiif, hal ini bergantung pada data dan permasalahan yang diselesaikan. CBOW menunjukkan kinerja yang unggul terhadap kata-kata yang sering muncul, sementara *Hierarchical Softmax* menampilkan hasil yang superior dengan pendekatan binary tree-nya, memungkinkan representasi vektor yang lebih baik karena cenderung mengabaikan kata-kata yang jarang muncul.

Dalam melakukan analisis sentimen mengenai ulasan obat (Colón-Ruiz & Segura-Bedmar, 2020) membandingkan dan mempelajari efek dari berbagai kombinasi arsitektur *Deep Learning* seperti CNN, LSTM, BiLSTM, hingga model *word embedding* BERT. Data yang digunakan merupakan data ulasan pada website Drugs.com sebanyak 215.063, yang kemudian dilakukan split data sebanyak 75% untuk *training* dan 25% untuk *testing*. Terdapat 2 skenario pembagian kelas pada penelitian ini, yaitu 10 kelas (penilaian 1 sampai 10) dan 3 kelas (positif, netral, dan negatif). Pada pembagian dataset 3 kelas, model BiLSTM-BERT mampu unggul namun membutuhkan waktu *training* dan biaya komputasi yang lebih tinggi,

sedangkan pada kondisi dataset 10 kelas model hybrid BILSTM-CNN yang lebih unggul. Disarankan untuk mengeksplorasi penerapan fitur semantik dengan fitur kontekstual sebagai upaya peningkatan pendekatan.

Penelitian (Zhao, Liu, Yao, & Yang, 2021) mengusulkan algoritma *machine learning* berbasis *neural network* yang dioptimalkan yaitu *Local Search Improvised Bat Algorithm based Elman Neural Network* (LSIBA-ENN). Algoritma ini diterapkan untuk melakukan analisis sentimen mengenai ulasan produk seperti aplikasi, hingga film dan TV yang diperoleh dari situs Amazon.com. Data ulasan aplikasi yang digunakan berjumlah 62.937 ulasan dan data film/TV sejumlah 1.677.539 ulasan. Algoritma ini dikombinasikan dengan beberapa *word embedding* yang ada seperti Word2Vec, TF, TF-IDF, TF-DFS, hingga *word embedding* yang dikembangkannya yaitu LTF-MICF. Algoritma LSIBA-ENN dengan fitur ekstraksi LTF-MICF mampu unggul dan mencapai efisiensi terkait matrik klasifikasinya, nilai akurasi yang diperoleh sebesar 92,01% untuk dataset aplikasi dan 93,01% untuk dataset film dan TV.

Penelitian (Widyaningrum & Kamayani, 2023) menganalisis sentimen opini masyarakat terhadap layanan transportasi *online* Maxim menggunakan algoritma *Naïve Bayes*. Data diambil dari Twitter dengan kata kunci "Maxim" dan dikategorikan menjadi sentimen positif dan negatif. Hasil evaluasi menunjukkan presisi sebesar 87% untuk sentimen negatif dan 80% untuk sentimen positif, serta recall sebesar 74% untuk sentimen negatif dan 90% untuk sentimen positif. Hasil analisis menunjukkan bahwa sebagian besar sentimen masyarakat terhadap layanan Maxim adalah positif, dengan 616 data positif dan 526 data negatif dari total 1170

tweets. Penelitian ini dapat dikembangkan dengan model analisis sentimen yang lebih canggih, seperti *deep learning* atau *neural network*.

Model *Long Short-Term Memory* (LSTM) dengan kombinasi Word2Vec dalam penelitian (Prawinata, Rahajoe, & Diyasa, 2024) diterapkan untuk melakukan analisis sentimen opini masyarakat terhadap kendaraan Listrik yang diungkapkan melalui situs twitter. Sebanyak 30.000 data opini diterapkan pada 4 skenario pengujian, yaitu Word2Vec dengan arsitektur Skip-gram dan CBOW, serta pembagian data 80:20 dan 70:30. Performa unggul diperoleh pada saat penerapan arsitektur Skip-gram dengan pembagian data 80:20, yaitu dengan akurasi sebesar 82.8%, presisi 82.68%, recall 82.8%, dan F1 Score 82.74%. Disarankan untuk mengembangkan model LSTM serta menerapkan teknik *oversampling* SMOTE untuk menghasilkan data penelitian dengan kelas data yang seimbang.

2.2. Keaslian Penelitian

Tabel 2. 1. Matriks literature review dan posisi penelitian

Analisis Penerapan Word Embedding Terhadap Bidirectional LSTM dalam Model Analisis Sentimen Ulasan Aplikasi

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	*Evaluation of Sentiment Analysis via Word Embedding and RNN Variants for Amazon Online Reviews*	Najla M. Alharbi, Norah S. Alghamdi, Eman H. Alkhammash, Jehad F. Al Amri Mathematical Problems in Engineering – Hindawi 2021	Tujuan dari penelitian ini adalah melakukan evaluasi terhadap berbagai variasi model *Deep Learning*, seperti RNN, LSTM-RNN, GLSTM-RNN, GRU-RNN, dan UG-RNN yang dikombinasikan dengan fitur ekstraksi berupa *word embedding* antara lain GloVe, Word2Vec, dan FastText dalam melakukan prediksi mengenai polaritas ulasan berupa teks sebanyak 400.000	Berdasarkan kombinasi algoritma dan ekstarksi fitur yang telah diuji coba, RNN berbasis GLSTM dengan kombinasi FastText mampu menghasilkan performa terbaik pada kondisi dataset tidak seimbang (93,75%). Hal ini berbeda pada kondisi dataset seimbang, performa terbaik dihasilkan oleh kombinasi LSTM-RNN dengan FastText (88,39%), lebih unggul 0,01% disbanding GLSTM. Sedangkan GloVe yang memiliki performa terbaik saat dikombinasikan dengan CNN justru	Kondisi data mempengaruhi hasil performa model, seperti jumlah data, data seimbang-tidak seimbang. Selain itu beberapa *word embedding* memiliki performa yang berbeda saat dikombinasikan dengan masing-masing algoritma, sehingga ekstraksi fitur dan *machine learning* dalam kasus analisis sentiment tidak dapat dinilai secara terpisah. Sehingga penerapan algoritma harus disesuaikan berdasarkan kebutuhan.	Penelitian selanjutnya akan menerapkan kombinasi algoritma Bi-LSTM dengan fitur ekstraksi *word embedding* Word2Vec, FastText, dan GloVe untuk data ulasan pengguna Google PlayStore.

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			ulasan pada ponsel dari Amazon.com.	memberikan performa terendah saat dikombinasikan dengan RNN.		
2	*GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Reviews*	Li Xiaoyan, Rodolfo C. Raga, Shi Xuemei Journal of Sensors – Hindawi 2022	Tujuan yang ingin dicapai dalam penelitian ini yaitu mengusulkan dan mengoptimalkan model analisis sentimen dengan kombinasi GloVe-CNN-BiLSTM terhadap komentar pada media sosial twitter mengenai COVID-19 sebanyak 81.696 komentar. Sebagai model *word embedding* GloVe digunakan untuk vektorisasi kata, CNN untuk merepresentasikan karakter, dan BiLSTM untuk.	Kombinasi GloVe-CNN-BiLSTM merupakan model yang paling unggul jika dibandingkan dengan model TextCNN dan model CNN-BiLSTM tanpa GloVe. Sehingga model tersebut secara efektif mampu melakukan identifikasi kecerderungan sentimental komentar pengguna, yakni dengan akurasi sebesar 0,9565 pada *complete-text-dataset*, 0,9509 pada *long-text-dataset*, dan 0,9560 pada *short-text-dataset*.	Penelitian ini hanya menggunakan satu jenis *word embedding* yaitu GloVe pada model GloVe-CNN-BiLSTM. Pada penelitian mendatang model tersebut akan diterapkan untuk analisis sentimen komentar berbahasa Mandarin dengan gabungan *long-text* dan *short-text*. Selain itu, model tersebut juga memiliki perluasan domain yang baik.	Penelitian yang akan dilakukan adalah mengusulkan model Bi-LSTM dengan kombinasi beberapa *word embedding*, seperti Word2Vec, FastText, dan GloVe.

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			membangun hubungan temporal.			
3	Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia	Dloifur Rohman Alghifari, Mohammad Edi, Lutfi Firmansyah Jurnal Manajemen Informatika (JAMIKA) 2021	Penelitian ini bertujuan untuk mengusulkan metode *Bidirectional* LSTM (Bi-LSTM) yang dikombinasikan dengan ekstraksi fitur dan pembobotan kata *Bag of Word* (BOW), serta menyajikan beberapa algoritma pembanding seperti *Multinomial Naïve bayes*, *Logistic Regression*, dan *Linear Support Vector Classifier* dalam melakukan analisis sentimen terhadap pelayanan Grab Indonesia	Bi-LSTM dengan kombinasi ekstraksi fitur *Bag of Word* (BOW) mampu diimplementasikan pada kasus analisis sentimen terhadap pelayanan Grab Indonesia dan memiliki performa yang paling unggul dibandingkan LSTM dan beberapa algoritma *machine learning* lainnya pada penelitian ini. Akurasi yang diperoleh Bi-LSTM sebesar 91% dnegan training loss sebesar 28%.	Bi-LSTM membutuhkan dataset dengan jumlah yang besar untuk menghindari *overfitting*, selain itu Bi-LSTM juga membutuhkan waktu dan biaya komputasi yang cukup tinggi. Sehingga disarankan penelitian selanjutnya dapat mempertimbangkan kombinasi *word embedding* agar dapat menghasilkan representasi kata yang lebih bervariasi dan banyak.	Pada penelitian selanjutnya akan mengusulkan metode Bi-LSTM yang dikombinasikan dengan beberapa jenis *word embedding*, seperti Word2Vec, FastText, dan GloVe. Serta menerapkan model pada dataset dengan jumlah lebih besar.

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			berdasarkan komentar pengguna aplikasi pada *playstore* yang berjumlah 5000 ulasan.			
4	Perbandingan Kinerja Word Embedding Word2Vec, GloVe, Dan FastText Pada Klasifikasi Teks	Arliyanti Nurdin, Bernadus Anggo Seno Aji, Anugrayani Bustamin, Zaenal Abidin Jurnal TEKNOKOMPAK 2020	Tujuan dalam penelitian ini adalah membandingkan kinerja word embedding dengan kombinasi algoritma *Convolutional Neural Network* dalam melakukan klasifikasi berita pada dataset 20 *Newsgroup* dan *Reuters Newswrite*. Adapun *word embedding* yang digunakan adalah Word2Vec, GloVe, dan FastText.	Dataset 20 *Newsgroup* terdiri atas 11.314 data latih dan 7.532 data uji, sedangkan dataset *Reuters Newswrite* terdiri atas 8.982 data latih dan 2.246 data uji. Kombinasi CNN + FastText memiliki performa yang unggul dibandingkan kombinasi CNN + Word2Vec dan CNN + GloVe. Nilai F-measure yang dihasilkan FastText, Word2Vec dan GloVe berturut-turut adalah 0.979, 0.925, dan 0.958 untuk dataset 20 *Newsgroup* dan 0.715,	Penggunaan setiap *word embedding* bergantung pada dataset yang digunakan dan permasalahan yang ingin diselesaikan	Penelitian selanjutnya tidak melakukan perbandingan penerapan *word embedding* Word2Vec, FastText, dan Glove pada CNN, melainkan membandingkan ketiga *word embedding* tersebut saat diterapkan pada Bi-LSTM.

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				0.694, dan 0.688 untuk dataset *Reuters*. Ketiga *word embedding* dinilai memiliki performa yang kompetitif karena perbedaan kinerja yang tidak begtiu signifikan.		
5	*Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text*	M. Alfa Riza, Novrido Charibaldi International Journal of Artificial Intelligence & Robotics (IJAIR) 2021	Penelitian ini bertujuan untuk menerapkan metode LSTM dengan *word embedding* FastText, serta memperbaiki performa Word2Vec, dan GloVe dalam melakukan deteksi emosi pada teks yang berasal dari twitter yang berjumlah 1304 data.	Penerapan Word2Vec dan FastText mendapatkan akurasi terbaik dalam penelitian ini, yakni sebesar 73,15%. Sedangkan penerapan GloVe menghasilkan akurasi sebesar 60,10%. FastText dinilai unggul karena mampu menangani masalah *out of vocabulary* (OOV).	Akurasi yang dihasilkan dinilai belum cukup baik sehingga disarankan penelitian selanjutnya dapat menerapkan metode *Deep Learning* lainnya, seperti CNN, BiLSTM, dan lainnya. Serta menggunakan data dengan jumlah yang lebih banyak.	Penelitian yang akan dilakukan menerapkan metode Bi-LSTM dengan kombinasi *word embedding* Word2Vec, FastText, dan GloVe untuk analisis sentimen ulasan pengguna pada situs Google Play Store. Serta menggunakan data dengan jumlah yang lebih banyak.
6	*Efficient Long Short-Term Memory-Based*	Naveen Kumar Gondhi, Chaahat,	Penelitian ini beertujuan untuk	Penerapan LSTM dengan *word*	Sebagai upaya dalam meningkatkan model	Penelitian selanjutnya akan menerapkan saran yang

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Sentiment Analysis of E-Commerce Reviews	Eishita Sharma, Amal H. Alharbi, Rohit Verma, Mohd Asif Shah Computational Intelligence and Neuroscience – Hindawi 2022	menggabungkan model *deep learning* LSTM dengan ekstraksi fitur *word embedding* Word2Vec dalam melakukan analisis sentimen dalam konteks ulasan *e-commerce* pada dataset Amazon 2018 yang berjumlah 938.261 ulasan.	*embedding* Word2Vec dalam dataset yang besar memperoleh hasil akurasi sebesar 89%, presisi 97%, recall 90%, dan f1-score 93%. Hasil ini menunjukan bahwa LSTM dengan ekstraksi fitur mampu unggul dibandingkan *Naïve Bayes*, *Support Classification*, hingga CNN dalam penelitian ini. Dengan kondisi dataset yang tidak siembang ukuran kinerja model yang digunakan adalah f1-score yaitu 93%.	disarankan untuk menerapkan LSTM dua arah untuk klasifikasi sentimen yang melatih dua rangkaian LSTM berurutan sesuai input aktual dan sebaliknya.	diberikan sebelumnya, yaitu untuk menerapkan model LSTM dua arah atau BiLSTM yang akan dikombinasikan dengan beberapa *word embedding*, seperti Word2Vec, FastText, dan GloVe dalam melakukan analisis sentimen terhadap suatu ulasan.
7	*An Automated Toxicity Classification on Social Media Using LSTM and Word Embedding*	Ahmad Alsharef, Karan Aggarwal, Sonia, Deepika Koundal, Hashem Alyami, Darine Ameyed	Penelitian ini dilakukan untuk melakukan peningkatan akurasi mengenai klasifikasi *toxicity* pada forum obrolan	Berdasarkan penerapan model LSTM dengan *word embedding* BERT pada kasus klasifikasi komentar (*toxic* dan *nontoxic*) diperoleh hasil akurasi sebesar	Dengan menerapkan teknik *AutoNLP* dan *AutoML* ke kumpulan data yang sama, model yang lebih kuat dapat dikembangkan karena Teknik ini mampu	Penelitian yang akan datang mengusulkan penerapan LSTM dua arah, yaitu BiLSTM yang dikombinasikan dengan *word embedding* Word2Vec, FastText, GloVe, dan BERT.

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Computational Intelligence and Neuroscience – Hindawi 2022	online dengan cara menerapkan *word embedding* BERT dan GloVe terhadap model LSTM.	94% dan F1-score sebesar 0,89. Sehingga dapat disimpulkan bahwa penambahan *word embedding* mampu meningkatkan keakuratan klasifikasi, kemudian kombinasi LSTM-BERT memiliki kinerja yang lebih baik dibandingkan LSTM tanpa *word embedding* dan LSTM-GloVe.	menemukan model yang paling sesuai berdasarkan datanya secara otomatis. Selain itu penelitian ini masih menggunakan LSTM dengan operasi urutan data satu arah.	
8	Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen	Dwi Intan Af'idah1, Dairoh, Sharfina Febbi Handayani, Riszki Wijayatun Pratiwi Jurnal Informatika: Jurnal Pengembangan IT (JPIT)2021	Penelitian ini mengusulkan pendekatan model analisis sentimen dengan menerapkan fitur ekstraksi berupa *word embedding* Word2Vec. Parameter Word2Vec seperti arsitektur (*Continuous Bag of Words*, *Skip-*	Parameter Word2vec berpengaruh terhadap kinerja *deep learning*. Berdasarkan parameter yang digunakan, kombinasi arsitektur CBOW, metode evaluasi *hierarchical softmax*, dan dimensi bernilai 100 menghasilkan akurasi yang unggul, meskipun performa tiap parameternya cukup	Model *word embedding* lain seperti FastText dan GloVe disarankan untuk diterapkan sebagai pembanding dari Word2Vec untuk mengetahui perbandingan performa model *word embedding* terhadap performa dari *deep learning*. Penerapan parameter *word embedding* bergantung pada data	Penelitian selanjutnya akan menerapkan parameter jumlah *window*, nilai dimensi dalam penerapan *word embedding* Word2Vec, selian itu kondisi dataset juga menjadi parameter lainnya. Kemudian *word embedding* lainnya seperti FastText dan GloVe juga diterapkan dalam penelitian yang akan dilakukan untuk klasifikasi analisis sentimen.

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			Gram), metode evaluasi (*hierarchical softmax, negative sampling*), dan nilai dimensi (100, 200, 300) diteliti untuk mengetahui pengaruh masing-masing penerapan parameter terhadap performa *deep learning*.	kompetitif. CBOW unggul dalam akurasi untuk kata-kata yang sering muncul, sementara *Hierarchical Softmax* lebih baik dalam merepresentasikan kata-kata yang jarang muncul dengan pendekatan *binary tree*-nya. Dalam dataset besar, *Hierarchical Softmax* lebih disukai karena mengabaikan kata-kata yang jarang muncul. Dimensi 100 dinilai optimal untuk dataset berjumlah 10.000 ulasan, sementara ukuran dimensi yang lebih besar dapat menurunkan kinerja model.	dan permasalahan yang akan diselesaikan.	

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
9	*Comparing deep learning architectures for sentiment analysis on drug reviews*	Cristóbal Colón-Ruiz, Isabel Segura-Bedmar Journal of Biomedical Informatics – Elsevier 2020	Penelitian ini bertujuan untuk menyajikan perbandingan berbagai arsitektur *Deep Learning* seperti *Convolutional Neural Network* dan *Long Short-Term Memory* beserta efeknya jika dikombinasikan dengan beberapa model *pre-trained word embedding*, seperti Bi-LSTM dengan BERT. Hal ini untuk melakukan analisis sentimen ulasan obat pada dataset yang diperoleh dari website Drugs.com yang berjumlah 215.063 ulasan.	Terdapat 2 macam pembagian kelas pada dataset penelitian ini, eksperimen dengan 3 kelas (positif, netral, negatif) memberikan haisl yang lebih tinggi dibandingkan ekperimen 10 kelas. Model *hybrid* BiLSTM-CNN unggul pada eksperimen 10 kelas dataset. Untuk eksperimen 3 kelas dataset, BiLSTM-BERT dengan waktu *training* dan biaya komputasi yang sangat tinggi mampu sedikit lebih unggul dibandingkan CNN.	BERT memerlukan waktu training dan biaya komputasi yang tinggi dibandingkan CNN. Eksplorasi metode baru untuk mengurangi ketergantungan serta eksplorasi penerapan fitur semantik dengan fitur kontekstual sebagai peningkatan pendekatan.	Penelitian yang akan dilakukan ingin menerapkan model dengan performa terbaik pada penelitian sebelumnya yakni Bi-LSTM dengan *word embedding* Word2Vec, FastText, dan GloVe untuk analisis sentimen ulasan pengguna Google Play Store.

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
10	*A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach*	Huiliang Zhao, Zhenghong Liu, Xuemei Yao, Qin Yang Information Processing and Management – Elsevier 2021	Penelitian ini mengusulkan algoritma *Machine Learning* (ML) baru berbasis *neural network* yang dioptimalkan yaitu *Local Search Improvised Bat Algorithm based Elman Neural Network* (LSIBA-ENN) untuk melakukan analisis sentimen terhadap ulasan produk online seperti aplikasi dan film pada website *e-commerce* (Amazon). Algoritma ini juga dikombinasikan degan fitur seleksi Word2Vec, TF,	Dataset yang digunakan merupakan data ulasan aplikasi yang berjumlah 62.937 ulasan dan data film dan TV yang berjumlah 1.677.539 ulasan. Algoritma LSIBA-ENN mampu memperoleh kinerja terbaik saat dikombinasikan dengan LTF-MICF pada 2 dataset yang digunakan dalam penelitian. LTF-MICF dinilai mampu mencapai efisiensi terkait matrik klasifikasinya. LSIBA-ENN dengan LTF-MICF mampu memeperoleh akurasi 92,01% pada dataset aplikasi dan 93,01% pada dataset film dan TV.	Model *machine learning* yang digunakan belum menerapkan ekstraksi fitur *word embedding*. Selain itu model dapat dikembangkan dengan menerapkan desain *deep learning* dalam melakukan analisis sentimen.	Penelitian yang akan dilakukan adalah mengusulkan algoritma *neural network* lainnya, yaitu *Bidirectional* LSTM yang dikombinasikan dengan ekstraksi fitur *word embedding* Word2Vec, FastText, dan GloVe untuk sentimen analisis ulasan pengguna Google Play Store.

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			TF-IDF, TF-DFS, dan LTF-MICF.			
11	Analisis Sentimen Opini Masyarakat Terhadap Penggunaan Layanan Maxim Menggunakan Algoritma Naïve Bayes	Intania Widyaningrum, Mia Kamayani Jurnal Computer Science and Information Technology 2023	Penelitian ini mengusulkan model analisis sentimen mengenai tanggapan masyarakat tentang jasa layanan transportasi online yaitu Maxim dengan menggunakan *Naïve Bayes* dimana proses pelabelan dilakukan dengan menggunakan kamus Leksikon Indonesia.	Berdasarkan data opini pengguna sebanyak 1.170 data yang dilabeli menggunakan kamus Leksikon Indonesia, model *Naïve Bayes* mampu menghasilkan performa akurasi sebesar 82,50%. Dalam penelitian ini juga diterapkan jenis *pre-processing* seperti *cleaning, case folding, stemming, tokenizing, dan remove stopword*.	Model yang terapkan menggunakan *machine learning* tanpa ekstraksi fitur maupun *word embedding*. Selain itu model dapat dikembangkan dengan menerapkan mesin yang lebih baik seperti *deep learning* atau *neural network* dalam melakukan analisis sentimen.	Penelitian yang akan dilakukan adalah mengusulkan algoritma *neural network*, yaitu Bidirectional LSTM yang dikombinasikan dengan ekstraksi fitur *word embedding* Word2Vec, FastText, dan GloVe untuk sentimen analisis ulasan pengguna Google Play Store.
12	Analisis Sentimen Kendaraan Listrik Pada Twitter Menggunakan	Dian Agus Prawinata, Ani Dijah Rahajoe, I Gede Susrama Mas Diyasa	Penelitian ini bertujuan untuk menganalisis opini mengenai penggunaan	Analisis sentimen dengan data berjumlah 30.000 opini yang diklasifikasikan dengan metode LSTM mampu	Perlu dilakukan eksplorasi model LSTM yang lebih kompleks untuk meningkatkan performa daya prediksi	Penelitian yang dilakukan menerapakan metode LSTM dua arah (*Bidirectional*) serta 3 jenis *word embedding* (Word2Vec, FastText, dan

Tabel 2. 1. Matriks literature review dan posisi penelitian (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Metode Long Short-Term Memory	SABER: Jurnal Teknik Informatika, Sains dan Ilmu Komunikasi 2024	kendaraan listrik dengan menggunakan metode klasifikasi *Long Short-Term Memory* (LSTM) dalam 4 (empat) skenario berbeda.	memperoleh performa terbaik pada skenario pembagian data 80:20 untuk data *training* dan data *testing*, serta pada saat menerapkan Word2Vec dengan arsitektur Skip-gram. Performa yang diperoleh antara lain akurasi sebesar 82.8%, presisi 82.68%, recall 82.8%, dan F1 Score 82.74%	untuk analisis sentimen berbahasa Indonesia. Selain itu juga perlu dilakukan penyeimbangan jumlah data untuk tiap kelasnya dengan menggunakan SMOTE.	GloVe) dengan kondisi data original (tidak seimbang) dan setelah *oversampling* dengan SMOTE.

2.3. Landasan Teori

2.3.1. Analisis Sentimen

Analisis sentimen adalah pendekatan yang menggunakan *Natural Language Processing* (NLP) (Khoo, et al., 2023) untuk secara otomatis menambang sikap, opini, perspektif, dan emosi dari teks, ucapan, tweet, dan sumber database. Dalam analisis sentimen, opini dalam sebuah teks dikategorikan ke dalam kategori "positif", "negatif", dan "netral". Istilah analisis subjektivitas, penambangan pendapat, dan ekstraksi penilaian juga digunakan untuk menggambarkannya (Kharde & Sonawane, 2016).

Analisis sentimen juga dikenal sebagai penambangan opini, istilah analisis sentimen mencakup berbagai aktivitas, antara lain, ekstraksi sentimen, klasifikasi sentimen, kategorisasi subjektivitas, ringkasan opini, dan deteksi spam opini yang dituangkan secara tertulis. Hal ini bertujuan untuk mengetahui bagaimana perasaan, sikap, dan pendapat seseorang mengenai berbagai faktor, termasuk perusahaan, orang, peristiwa, masalah, barang, subjek, organisasi, hingga pelayanan. Ini juga bertujuan untuk memeriksa sikap dan pendapat mereka (Bhatia, Sharma, & Bhatia, Sentiment Analysis and Mining of Opinions, 2017).

Meskipun istilah "pendapat", "sentimen", "pandangan", dan "kepercayaan" sering digunakan secara bergantian, mereka memiliki arti yang berbeda. Opini: Suatu penilaian yang tunduk pada ketidaksepakatan (karena penilaian berbagai ahli berbeda), Pandangan: Penilaian pribadi, Keyakinan: secara sadar mengekspresikan persetujuan dan persetujuan pikiran, dan Sentimen: Pendapat yang mengekspresikan emosi seseorang (Kharde & Sonawane, 2016)

Analisis sentimen, juga disebut penambangan opini, adalah bidang studi yang menganalisis opini, sentimen, penilaian, sikap, dan emosi orang terhadap entitas dan atributnya yang diungkapkan dalam teks tertulis. Entitas dapat berupa produk, layanan, organisasi, individu, acara, isu, atau topik. Bidang mewakili ruang masalah yang besar. Banyak nama terkait dan tugas yang sedikit berbeda, misalnya, analisis sentimen, penambangan opini, analisis opini, ekstraksi opini, penambangan sentimen, analisis subjektivitas, analisis pengaruh, analisis emosi, dan penambangan ulasan - kini semuanya berada di bawah payung analisis sentimen (Liu B. , 2020).

2.3.2. Word Embedding

Word embedding menggunakan representasi numerik dan vektor dari setiap kata dengan mengkonversi kata dasar kedalam bentuk vector (Poria, Hussain, & Cambria, 2018). *Word embedding* menjelaskan teks yang berisi replika tepat dari kata-kata yang memiliki arti yang sama. Penyematan kata khususnya adalah pembelajaran representasi kata tanpa pengawasan yang sebanding dengan kesamaan semantik. Dalam skema terkoordinasi, frase yang sebanding dikelompokkan bersama lebih dekat bersama berdasarkan serangkaian hubungan (Iqbal, et al., 2022). *Word embedding* termasuk dalam fitur ekstraksi, beberapa *word embedding* mengusulkan konsep atau Teknik yang serupa (Wang, Wang, Chen, Wang, & Kuo, 2019).

Word embedding atau representasi vektor dari kata-kata, merupakan komponen penting dari perangkat *Natural Language Processing* (NLP) (Mikolov,

Sutskever, Chen, Corrado, & Dean, 2013) (Pennington, Socher, & Manning, 2014). Secara teoritis, merepresentasikan kata-kata sebagai vektor daripada variabel diskrit memungkinkan untuk generalisasi di kata-kata yang mirip secara sintaksis atau semantik, dan algoritma penyisipan kata yang mudah diterapkan dan dilatih telah membuat penyisipan kata berkualitas tinggi tersedia untuk sebagian besar domain dan Bahasa (Søgaard, Vulić, Ruder, & Faruqui, 2022).

Adapun *Word embedding* yang diterapkan pada penelitian ini antara lain Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), FastText (Bojanowski, Grave, & Mikolov, 2017), GloVe (Pennington, Socher, & Manning, 2014), dan BERT (Devlin, Chang, Lee, & Toutanova, 2019).

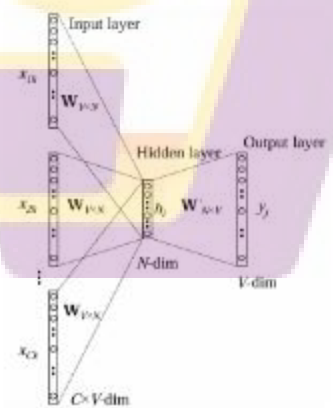
2.3.2.1 Word2Vec

Word2Vec adalah salah satu metode penyisipan kata yang diperkenalkan oleh Mikolov et al. pada tahun 2013. Dengan berkonsentrasi pada kata yang dekat dengan kata target, Word2Vec mampu menemukan arti kata yang mirip dengan kata target. Karena kelebihannya, metode Word2Vec sangat populer. Word2Vec menggunakan dua teknik: metodologi *Continuous Bag of Word* (CBOW) dan model *skip-gram* (Riza & Charibaldi, 2021). Word2vec mengubah kata-kata menjadi vektor yang menyimpan makna semantik dari kata tersebut. Model *word embedding* ini merupakan salah satu contoh aplikasi dari *unsupervised learning* yang menggunakan *neural network*. Model tersebut terdiri dari lapisan tersembunyi (*hidden layer*) dan lapisan *fully connected*.

Dimensi dari matriks bobot pada setiap lapisan ditentukan oleh jumlah kata dalam korpus dikalikan dengan jumlah *hidden neuron* pada lapisan tersembunyi. Matriks bobot pada lapisan tersembunyi yang telah dilatih digunakan untuk mentransformasikan kata-kata menjadi vektor. Matriks bobot ini berfungsi seperti tabel pencarian (*lookup table*), di mana setiap baris merepresentasikan satu kata dan setiap kolom merepresentasikan vektor dari kata tersebut. Dengan menggunakan matriks bobot ini, kata-kata dapat direpresentasikan sebagai vektor dalam ruang semantik (Nurdin, Aji, Bustamin, & Abidin, 2020).

CBOW

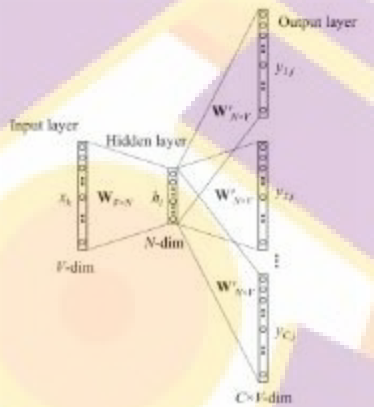
Pendekatan ini memprediksi kata target berdasarkan konteks (Gambar 2.1). Untuk istilah yang sering muncul, CBOW menunjukkan akurasi yang sedikit lebih besar dan waktu pelatihan yang lebih cepat (Nurdin, Aji, Bustamin, & Abidin, 2020).



Gambar 2. 1. Arsitektur CBOW

Skip-gram

Berbeda dengan CBOW, Skip-gram justru menggunakan kata target untuk memprediksi kata konteks disekitarnya (Gambar 2.2.). Skip-gram dinilai mampu bekerja dengan baik pada kondisi data pelatihan dengan jumlah yang kecil karena mampu mewakili kata-kata yang memiliki frekuensi kemunculan sedikit (Nurdin, Aji, Bustamin, & Abidin, 2020).



Gambar 2. 2. Arsitektur Skip-gram

2.3.2.2 FastText

FastText adalah teknik *word embedding* yang berevolusi dari Word2Vec. Dengan mempertimbangkan informasi subkata, pendekatan ini menyelidiki representasi kata. Sekelompok karakter *n-gram* berfungsi sebagai representasi untuk setiap kata. Hal ini memungkinkan penyematan untuk memahami akhiran dan awalan kata dan dapat membantu dalam menangkap arti dari kata-kata yang lebih pendek. Setiap huruf *n-gram* memiliki representasi vektor yang terkait

dengannya, dan kata-kata hanyalah total dari semua representasi vektor tersebut. Setelah kata dikodekan sebagai karakter *n-gram*, model *Skip-gram* dilatih untuk menemukan vektor penyisipan kata (Nurdin, Aji, Bustamin, & Abidin, 2020).

Setiap kata yang direpresentasikan sebagai sekumpulan karakter n-gram, disimbolkan dengan < dan > pada bagian awal dan akhir kata, ini juga berfungsi untuk membedakan prefix dan sufiks dari rangkaian karakter lainnya. Seperti halnya untuk kata "belanja" dengan n-gram=3, kata tersebut akan diwakili oleh karakter n-gram: <bel, ela, lan, anj, nja> dan barisan khusus <belanja>. Urutan kata <bel> pada kata "bel" memiliki perbedaan dengan 3-gram bel yang berasal dari kata "belanja". Selain itu, menurut (Bojanowski, Grave, & Mikolov, 2017), Softmax yang digunakan dalam model Skip-gram menghasilkan kapasitas prediksinya yang terbatas yakni hanya dalam satu konteks. Sehingga diusulkan pendekatan FastText yang direpresentasikan pada Persamaan 1.

$$\sum_{t=1}^{T}\left[\sum_{c\epsilon C_t}\ell(s(w_t,w_c))+\sum_{n\epsilon N_{t,c}}\sum_{c\epsilon C_t}\ell(-s(w_t,n))\right] \quad (1)$$

Dimana:

ℓ = log (1+e-x)

S = *scoring function*

w = weight

n = jumlah kosa kata (*vocabulary*)

Morfologi kata biasanya diabaikan oleh model yang menganalisis representasi kata sebagai vektor; sebaliknya, setiap kata memiliki vektor unik. Pembatasan ini berlaku untuk bahasa dengan repertoar kata yang luas dan jumlah kata yang tidak biasa. FastText menawarkan representasi untuk kata-kata yang tidak

muncul dalam data pelatihan, memberikan kinerja yang baik, dan melatih model dengan cepat pada kumpulan data yang besar. Kata tersebut dapat dipecah menjadi *n-gram* untuk mendapatkan vektor penyisipan jika tidak terjadi selama pelatihan model (Nurdin, Aji, Bustamin, & Abidin, 2020).

FastText menghasilkan penyematan kata untuk teks yang masuk. FastText membuat model ruang vektor dimensi tinggi selama pelatihan yang bertujuan untuk menangkap makna sebanyak mungkin dengan menganalisis korpus teks masukan. Ruang vektor dirancang dengan gagasan bahwa vektor kata terkait harus berdekatan satu sama lain. Vektor kata ini kemudian disimpan dalam FastText (Bhattacharjee, 2018).

2.3.2.3 GloVe

Algoritma GloVe mengintegrasikan informasi *co-occurrence* kata atau statistik global untuk memperoleh asosiasi semantik antara kata-kata dalam korpus (Nurdin, Aji, Bustamin, & Abidin, 2020), berbeda dengan Word2Vec, yang secara eksklusif bergantung pada informasi lokal dari kata-kata dengan *local context window* (CBOW dan *Skip-gram*). Pendekatan *global matrix factorization*, yang digunakan GloVe, membuat matriks yang menggambarkan keberadaan atau kekurangan kata dalam dokumen. Algoritma GloVe diperkenalkan oleh (Pennington, Socher, & Manning, 2014) yang dinyatakan dalam Persamaan 2.

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \bar{w}_j + b_i + \bar{b}_j - logX_{ij})^2 \qquad (2)$$

Dimana:

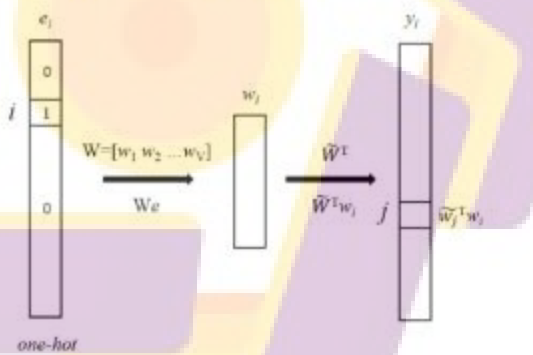
w_i dan $\bar{w}_j$ merupakan vector kata untuk i dan kata konteks j

X_{ij} merupakan matriks *co-occurrence* yang dibobotkan

V merupakan ukuran kosa kata

b_i dan $\tilde{b}_j$ merupakan bias untuk kata i dan kata konteks j

Model GloVe adalah tools yang berguna untuk memanfaatkan data korpus global dan mengoptimalkan model pembelajaran sesuai dengan jendela konteks. Tujuan utamanya adalah untuk membuat vektor kata dan menghasilkan vektor kata menggunakan korpus input. Prosedur pelaksanaannya adalah sebagai berikut: pertama, buat matriks *co-occurence* kata menggunakan korpus lengkap, kemudian mengolah vektor kata pembelajaran menggunakan matriks *co-occurrence* dan model GloVe (Xiaoyan, Raga, & Xuemei, 2022). Dimana menurut (Liu, Zhang, Li, & Yan, 2019) arsitektur GloVe diilustrasikan seperti Gambar 2.3.



Gambar 2. 3. Arsitektur GloVe (Liu, Zhang, Li, & Yan, 2019)

2.3.3. Preprocessing

Preprocessing merupakan tahapan dimana data melalui persiapan sebelum data tersebut digunakan pada model untuk proses klasifikasi (Bhattacharjee, 2018).

Adapun *preprocessing* yang akan dilakukan pada penelitian ini adalah *clean text*, *tokenization*, *stemming*, dan *stop word removal*.

2.3.3.1 Clean Text

Ulasan pelanggan ditulis tanpa memperhatikan tata bahasa atau norma penulisan teks, teks yang dimasukkan dapat mengandung huruf kecil-huruf kapital, tanda baca, angka-angka, hingga *emoticon*, hal ini berakibat pada sulitnya pengklasifikasi dalam menentukan polaritas teks (Iqbal, et al., 2022). *Clean Text* merupakan tahapan untuk menyeragamkan bentuk teks dan membersihkan data teks dari data yang tidak digunakan dalam pemrosesan selanjutnya, seperti menghilangkan atau menghapus tanda baca, angka, simbol-simbol, *hyperlink*, *hash tag* (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) (Riza & Charibaldi, 2021), hingga merubah data teks menjadi *lower case* (Kharde & Sonawane, 2016) (Nurdeni, Budi, & Santoso, 2021) (Iqbal, et al., 2022).

2.3.3.2 Tokenization

Tokenization atau tokenisasi merupakan proses mengidentifikasi esensi karakter dalam teks dengan menekankan kata kunci dan membuat titik di mana satu kata muncul terlebih dahulu dan kata lainnya mengikuti. Untuk tujuan komputasi linguistik, kata-kata yang diidentifikasi dengan cara ini sering disebut sebagai token. Tokenisasi juga dikenal sebagai segmentasi kata dalam bahasa yang tidak memiliki batas kata yang ditransmisikan secara elaboratif saat menulis, dan istilah ini sering digunakan bersamaan dengan tokenisasi (Dale, Moisi, & Somers, 2000).

Tokenisasi adalah proses pengenalan kata-kata dalam urutan input karakter, terutama dengan memisahkan tanda baca, tetapi juga dengan pengenalan, singkatan, dll. Dalam beberapa kasus, jika konten diambil dari web halaman, proses tokenisasi juga mencakup prosedur untuk menormalkan teks, seperti menghapus Tag HTML, *truecasing*, atau huruf kecil (Zhao, Liu, Yao, & Yang, 2021). *Tokenization* memecah dokumen teks menjadi token-token berdasarkan karakter spasi, hal ini untuk mempermudah model NPL dalam memahami konteks dan menganalisis urutan kata dalam memberikan makna teks tersebut (Aljedaani, Rustam, Ludi, Ouni, & Mkaouer, 2021) (Zhao, Liu, Yao, & Yang, 2021).

2.3.3.3 Stemming

Banyak kata dalam bahasa alami yang memiliki kesamaan dalam penulisan, selain itu keberagaman bentuk kata membuat pengakuan tidak berguna atau ambigu. Sederhananya, *stemming* adalah proses transformasi kata dengan menghilangkan imbuhan seperti prefiks, sufiks dan sisipan dari kata input untuk mendapatkan kata dasar (Riza & Charibaldi, 2021). Tujuan *stemming* adalah untuk mengurangi varian kata-kata yang memiliki arti yang hampir sama dalam sebuah dokumen guna mempermudah dan meningkatkan kemampuan proses pencarian informasi (Aljedaani, Rustam, Ludi, Ouni, & Mkaouer, 2021).

2.3.3.4 Stopword Removal

Stop word removal merupakan proses dimana kata-kata yang tidak memiliki kontribusi pada dokumen akan dihilangkan (Aljedaani, Rustam, Ludi, Ouni, &

Mkaouer, 2021). *Stop word* diartikan sebgaai kata-kata yang berfrekuensi tinggi, seperti *pronouns*, *determinans*, *prepositions*, dan lainnya. File teks sering berisi kata-kata yang diulang. Oleh karena itu, sangat penting untuk menghapus *stopword*. Pada kenyataannya, *stopword* tidak pernah memberikan makna apa pun pada konten tertulis. Tentu hal ini dapat mempersulit proses *text mining* dan memberikan hasil klasifikasi yang buruk. Sehingga dengan *stopword removal* diharapkan mamppu meningkatkan efisiensi model (Iqbal, et al., 2022). Meskipun demikian *stop word* mampu meningkatkan performa model (Bhattacharjee, 2018).

2.3.4. SMOTE

Teknik pengambilan sampel berlebihan yang melibatkan pembuatan contoh "sintetis" daripada mengganti data untuk mengambil sampel berlebihan pada kelas minoritas. Untuk menghasilkan data pelatihan tambahan, mereka menggunakan aktivitas tertentu pada data aktual. Setelah mengambil sampel dari setiap kelas minoritas, contoh sintetik ditambahkan di sepanjang garis yang menghubungkan salah satu atau semua tetangga terdekat dari kelas minoritas, sehingga terjadi *oversampling* pada kelas minoritas. Bergantung pada tingkat pengambilan sampel berlebih yang diperlukan, tetangga yang dipilih secara acak dipilih di antara tetangga terdekat. Misalnya, jika diperlukan pengambilan sampel berlebih sebesar 200%, maka hanya dua tetangga yang dipilih dari lima tetangga terdekat, dan satu sampel dihasilkan di setiap cara. Proses pembuatan sampel sintetik adalah sebagai berikut: Menentukan selisih antara vektor ciri (sampel yang diteliti) dengan tetangga terdekatnya. Lipat gandakan perbedaan ini dengan angka acak antara 0 dan

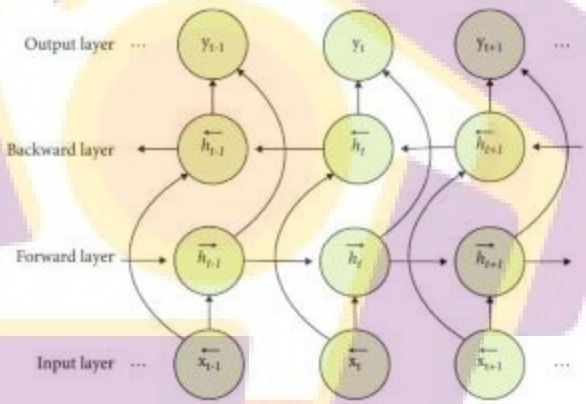
1 untuk menambahkannya ke vektor fitur yang sedang diperiksa. Hasilnya, ruas garis yang menghubungkan dua fitur tertentu dipilih secara acak sepanjang garis tersebut. Taktik ini pada dasarnya memaksa wilayah pengambilan keputusan kelas minoritas untuk memperluas wawasannya (Chawla, Bowyer, Hall, & Kagelmeyer, 2002).

2.3.5. Bidirectional Long Short-Term Memory

Pada model *recurrent neural network* tradisional dan LSTM, informasi hanya disebarkan searah yaitu ke arah depan. Agar setiap momen mengandung informasi konteks, BiLSTM yang menggabungkan model *bidirectional recurrent neural network* (BiRNN) dan unit LSTM digunakan untuk menangkap informasi konteks. Model BiLSTM memperlakukan semua input secara setara, tetapi dalam analisis sentimen, polaritas sentimen teks sangat bergantung pada kata-kata dengan informasi sentimen. Oleh karena itu, metode ini mengenalkan vektor kata sentimen yang diperkuat untuk mengatasi masalah ini (Xu, Meng, Qiu, Yu, & Wu, 2019).

Pertama, vektor kata terbobot yang menggabungkan informasi sentimen dan kontribusi klasifikasi dibangun. Vektor kata ini kemudian digunakan sebagai masukan untuk model BiLSTM. Keluaran dari model BiLSTM digunakan sebagai representasi teks komentar. Selanjutnya, vektor representasi teks komentar dimasukkan ke dalam *feedforward neural network classifier*. Fungsi aktivasi yang digunakan dalam jaringan saraf umpan maju adalah fungsi ReLU. Untuk mencegah *overfitting* selama proses pelatihan, mekanisme *dropout* diperkenalkan dengan tingkat *dropout* sebesar 0,5 (Xu, Meng, Qiu, Yu, & Wu, 2019).

Metode ini menggambarkan diagram skematik yang terdiri dari dua subgraf. Subgraf kiri menggambarkan proses ekstraksi fitur dari teks komentar menggunakan vektor kata terbobot dan model BiLSTM. Subgraf kanan menggambarkan proses untuk mendapatkan polaritas sentimen dari teks komentar menggunakan *feedforward neural network classifier*. Jumlah node dalam lapisan tersembunyi LSTM ditunjukkan oleh NodeNum. Metode ini bertujuan untuk meningkatkan kinerja analisis sentimen dengan mempertimbangkan kontribusi kata-kata dengan informasi sentimen dalam tugas kategorisasi teks. Arsitektur BiLSTM dapat dilihat pada Gambar 1.



Gambar 2. 4. Bidirectional LSTM (Xiaoyan, Raga, & Xuemei, 2022)

2.3.6. Confusion Matrix

Confusion Matrix adalah matriks yang menunjukkan seberapa baik model meramalkan keadaan tertentu. *Confusion matrix* adalah salah satu alat untuk menilai kinerja model klasifikasi (Iqbal, et al., 2022). Memanfaatkan sejumlah pendekatan evaluasi, kemanjuran klasifikasi pada data uji yang belum terlihat

dievaluasi (Haddi, Liu, & Shi, 2013). Metrik yang paling sering digunakan untuk klasifikasi teks termasuk presisi, *recall, F-measure*, dan akurasi. Tujuannya adalah untuk mengoptimalkan semua ukuran, yang memiliki nilai antara 0 dan 1. Oleh karena itu, angka yang lebih tinggi menunjukkan kinerja kategorisasi yang lebih baik.

Confussion Matrix digunakan untuk melihat bagaimana kinerja model terkait dengan berbagai label. Confussion Matrix memberikan pemahaman yang baik tentang *true negative* (TN) dan *false positive* (FP), *false positive* (FP), dan *false negative* (FN) (Bhattacharjee, 2018).

Tabel 2. 2. Confusion Matrix

Confusion Matrix		**Prediksi**	
		Positif	**Negatif**
Aktual	**Positif**	TP (*True Positive*)	FN (*False Negatice*)
	Negatif	FP (*False Negative*)	TN (*True Negative*)

Parameter perhitungan didefinikan sebagai berikut (Yang, Li, Wang, & Siherratt, 2020) (Subba & Kumari, 2022).

TP : Jumlah ulasan yang memiliki sentimen positif yang diklasifikasikan sebagai ulasan positif.

FP : Jumlah ulasan yang memiliki sentimen negatif, namun diklasifikasikan sebagai ulasan positif.

TN : Jumlah ulasan yang memiliki sentimen negatif yang diklasifikasikan sebagai ulasan negatif.

FN : Jumlah ulasan yang memiliki sentimen positif, namun diklasifikasikan sebagai ulasan negatif

Akurasi: Rasio ulasan yang diklasifikan dengan benar sesuai data *actual* terhadap jumlah ulasan seluruhnya.

$$Akurasi = \frac{TP+TN}{TP+TN+FN+FP} \times 100\% \quad (3)$$

Akurasi untuk klasifikasi *multi class* didefinisikan dalam persamaan berikut.

$$Akurasi = \frac{Jumlah\ prediksi\ yang\ benar}{Jumlah\ prediksi\ seluruhnya} \times 100 \quad (4)$$

Presisi: Rasio ulasan yang memiliki sentiment positif yang diklasifikan dengan benar terhadap jumlah seluruh review yang diklasifikasikan sebagai ulasan positif.

$$Presisi = \frac{TP}{TP+FP} \quad (5)$$

Presisi untuk klasifikasi *multi class* didefinisikan dalam persamaan berikut.

$$Presisi = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \quad (6)$$

Recall: Rasio ulasan yang memiliki sentiment positif yang diklasifikan dengan benar terhadap jumlah seluruh ulasan yang aktualnya merupakan review positif.

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

Recall untuk klasifikasi *multi class* didefinisikan dalam persamaan berikut.

$$Recall = \frac{\sum_{i=1}^{n} FP_i}{\sum_{i=1}^{n} FP_i + \sum_{i=1}^{n} TN_i} \quad (8)$$

F1-Measure: Nilai rata-rata tertimbang dari presisi dan recall.

$$F1 - Measure = \frac{2 \times Recall \times Presisi}{Recall + Presisi} \quad (9)$$

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

3.1.1. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian eksperimen. Eksprerimen dalam penelitian ini dilakukan untuk mengetahui pengaruh dan performa penerapan *word embedding* Word2Vec, FastText, dan GloVe saat diterapkan pada algoritma Bi-LSTM dalam melakukan analisis sentimen terhadap ulasan pengguna Google PlayStore terhadap aplikasi Gojek.

3.1.2. Sifat Penelitian

Penelitian yang dilakukan bersifat deskriptif. Menggambarkan ulasan-ulasan yang diberikan pengguna dalam melakukan ulasan terhadap aplikasi Gojek pada situs Google Play Store. Serta akan dijabarkan hasil-hasil pengujian yang diperoleh dalam penelitian ini mengenai performa (akurasi, presisi, recall, f1-score) setiap penerapan *word embedding*.

3.1.3. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan data berupa ulasan terhadap aplikasi Gojek yang diberikan oleh pengguna situs Google PlayStore. Nantinya hasil dalam penelitian ini berupa angka yang disajikan dengan grafik dan tabel.

3.2. Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini menggunakan data ulasan yang diungkapkan oleh pengguna pada situs Google PlayStore mengenai aplikasi Gojek. Adapun data merupakan dataset publik yang tersedia dan dapat diperoleh melalui situs penyedia dataset.

Dataset ini dipilih karena merupakan ulasan aplikasi Gojek yang diungkapkan oleh pengguna aplikasi pada situs Google PlayStore. Aplikasi Gojek merupakan *decacorn* buatan Indonesia yang telah diunduh lebih dari 100.000.000 kali. Google Playstore juga dianggap sebagai sumber terbaik untuk menemukan kumpulan data bersih untuk penelitian analisis sentimen karena berisi ungkapan pengguna tanpa konten komersial atau promosi (Muttaqin & Khasirudin, 2021). Jumlah data yang diperoleh dan digunakan dalam penelitian ini berjumlah 44.917 data ulasan.

3.3. Metode Analisis Data

Pengembangan metode yang dilakukan dalam penelitian ini yaitu dengan menerapkan *word embedding* Word2Vec, FastText, dan GloVe terhadap metode Bi-LSTM. Berikut merupakan langkah-langkah analisis data dalam penelitian ini.

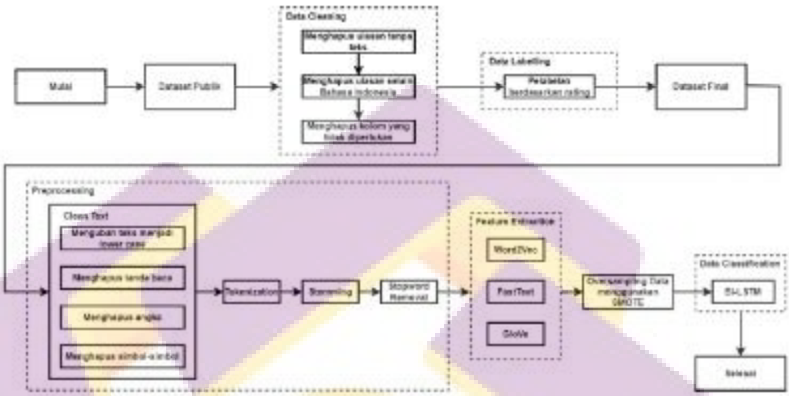
a. *Cleaning Data,* merupakan tahap pertama pengolahan teks yang dilakukan terhadap sumber data dalam penelitian. Tahapan ini terdiri atas beberapa proses, yaitu mem*filter* ulasan yang tidak memiliki teks ulasan, kemudian mem*filter* ulasan yang diungkapkan menggunakan Bahasa selain Bahasa Indonesia, dan menghapus kolom-kolom data yang tidak diperlukan. Dari

tahapan ini diperoleh dataset final yang digunakan untuk tahapan selanjutnya.

b. *Labelling*, proses pemberian label untuk setiap data teks ulasan dilakukan berdasarkan *rating* pada situs Google Playstore. Ulasan dengan *rating* 1-3 dilabeli sebagai sentimen Negatif, dan ulasan dengan *rating* 4-5 dilabeli sebagai sentimen Positif.
c. *Text Preprocessing*, dataset final yang akan diklasifikasikan akan melalui tahapan *preprocessing* terlebih dahulu. Adapun jenis *preprocessing* yang digunakan adalah *Clean Text, Tokenization, Stemming,* dan *Stop word Removal*.
d. *Feature Extraction*, tahapan ini merupakan penerapan fitur-fitur yang akan digunakan dalam analisis sentimen. Fitur ekstraksi yang digunakan yaitu *word embedding* Word2Vec, FastText, dan GloVe.
e. *Oversampling Data,* dilakukan untuk menangani data dengan kondisi yang tidak seimbang. Tahapan ini dilakukan dengan menerapkan metode SMOTE.
f. *Classifier*, tahapan ini digunakan untuk menentukkan apakah ulasan pelanggan cenderung bersifat positif atau negatif. Metode yang digunakan untuk tahapan klasifikasi adalah Bidirectional LSTM.

Beberapa tahapan yang perlu dilakukan terhadap data untuk analisis sentimen pada penelitian ini antara lain proses pengumpulan data ulasan pengguna, dimana dalam penelitian ini data yang digunakan merupakan dataset yang bersifat

publik, selanjutnya data dibersihkan, melalui tahapan preprocessing, *oversampling*, hingga klasifikasi. Adapun alur pemrosesan datanya dapat dilihat pada Gambar 3.1.



Gambar 3. 1. Alur Pemrosesan Data

3.4. Alur Penelitian

Penelitian ini terdiri atas beberapa tahapan seperti yang diilustrasikan pada Gambar 3.2. Diawali dengan tahap pengumpulan data (*data collecting*), tahap membersihkan data (*data cleaning*/*data filtering*), tahap pelabelan data (*data* labelling), tahap prapemrosesan data (*data preprocessing*), tahap pembagian dataset (*split dataset*), tahap pembobotan dengan ekstraksi fitur, tahap *oversampling* data, tahap klasifikasi atau analisis sentimen, sampai dengan tahap evaluasi dan analisis hasil evaluasi. Adapun untuk penjelasan alur pada penelitian ini adalah sebagai berikut.

1. *Data Collection*

   Kumpulan data yang digunakan dalam penelitian ini merupakan data ulasan yang diungkapkan para pengguna aplikasi pada situs Google

Playstore, data tersebut dapat diakses secara publik. Data yang digunakan merupakan ulasan dari aplikasi Gojek. Jumlah data yang digunakan mengacu pada penelitian (Alghifari, Edi, & Firmansyah, 2022) dan (Riza & Charibaldi, 2021) yang menyarankan untuk menggunakan dataset dengan jumlah yang lebih banyak untuk menangani permasalahan *overfitting*, yakni lebih dari 5.000 data.

2. *Data Cleaning*

Data ulasan yang digunakan akan dibersihkan dari data-data yang tidak diperlukan untuk pemrosesan selanjutnya. Pembersihan data dilakukan dengan cara sebagai berikut.

- Menghapus data ulasan yang tidak memiliki teks ulasan.
- Menghapus data ulasan yang berbahasa selain Bahasa Indonesia.
- Menghapus kolom yang tidak diperlukan pada data ulasan.

Hasil pada tahapan ini merupakan dataset final yang akan digunakan untuk tahapan selanjutnya dalam analisis sentimen

3. *Data Labelling*

Data yang sudah dibersihkan akan melalui tahapan pelabelan. Proses pelabelan data dilakukan berdasarkan rating ulasan. Data ulasan dengan *rating* 1-3 dilabeli sebagai sentimen Negatif, dan ulasan dengan *rating* 4-5 dilabeli sebagai sentimen Positif. Metode pelabelan seperti ini juga diterapkan pada beberapa penelitian (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) (Yang, Li, Wang, & Siherratt, 2020). Berikut disajikan contoh penentuan label untuk teks ulasan pada Tabel 3.1.

Tabel 3. 1. Pelabelan Data Berdasarkan Rating

Content	Rating	Label
kode otp gak masuk masuk, udah update terbaru, udah clear cache, tetep gak bisa masuk, udah on-off airplane tetep gak masuk, padahal sinyal full, udah coba kirim ulang berkali kali juga gak masuk, jadi gak bisa top up dengan oneklik bca.	★☆☆☆☆	Negatif
Dulu: Banyak bonus. Bisa pesan lintas kota. kurangnya cuma yg super partner, klo menu ga sesuai, ga bisa di cancel walaupun resto dah konfirmasi. Sekarang: "Lebih mahal ongkos kirim dan biaya lain-lain" daripada harga satuan barang/makanan tsb. Ga logis. Dan ga ada diskon walau beli banyak s.d 200rban, klo mau ada diskon harus beli voucher dulu.	★★★☆☆	Negatif
Driver2nya bs antar dg tepat, ga bingung arah dan tdk salah pesanan, tdk spt pengalaman saya di aplikasi lainnya. User friendly dan lengkap. Go pay jg sgt membantu. Skrg driver2nya sangat responsif saat hujan, terimaksh masukan saya didengarkan. Mode hemat jg sangat membantu saat kantong tipis.	★★★★★	Positif
Akan lebih baik jika lokasi tidak disetting otomatis di lokasi sekitar. Boleh jadi, pelanggan ingin mengecek biaya gocar/goride suatu tempat dari jarak jauh. Misal, bandara ke hotel. Setting lokasi sekitar mengakibatkan pengecekan biaya tak dapat dilakukan. Sebab, lokasi yang jauh tak terdeteksi.	★★☆☆☆	Negatif
Performa makin lama makin menurun dan harga yg di tawarkan juga lebih tinggi dibanding aplikasi sebelah coba diperbaiki sistemnya lagi seperti awal jangan kebanyakan upgrade aplikasi makin lemot dan berat keseringan di perbaharui.	★☆☆☆☆	Negatif

4. *Data Preprocessing*

Selanjutnya dataset final akan mengalami tahapan *preprocessing*. Data ulasan tentunya diungkapkan oleh orang yang berbeda dan dengan cara yang berbeda sehingga memiliki polaritas yang sangat rentan terhadap inkonsistensi dan redundansi, maka diperlukan tahapan preprocessing yang bertujuan untuk mempermudah pengamatan dan meningkatkan algoritma

yang diusulkan dalam melakukan analisis sentimen (Aljedaani, Rustam, Ludi, Ouni, & Mkaouer, 2021) (Kharde & Sonawane, 2016).

Preprocessing yang digunakan dalam penelitian ini antara lain *Clean Text*, *Tokenization*, *Stemming* dengan Sastrawi, dan *Stopword Removal*. *Tokenization* berfungsi untuk membagi setiap kata menjadi token, *Stemming* berfungsi untuk menghilangkan kata imbuhan agar setiap kata dasar yang sama memiliki bentuk kata yang seragam meskipun sebelumnya memiliki bentuk imbuhan yang berbeda-beda, dan *Stopword Removal* untuk menghilangkan tanda baca seperti titik, koma, serta menghilangkan kata dasar yang tidak memiliki makna (Alghifari, Edi, & Firmansyah, 2022) (Aljedaani, Rustam, Ludi, Ouni, & Mkaouer, 2021) (Cahyo, et al., 2023) (Rahman, Utami, & Sudarmawan, 2021).

5. *Split Data*

Pada tahapan ini data akan dibagi menjadi *Training Data* dan *Testing Data* dengan perbandingan 80:20. Pembagian data ini berdasarkan beberapa penelitian lainnya (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) (Cahyo, et al., 2023) (Haddi, Liu, & Shi, 2013) (Putri & Khasirudin, 2022) (Xiaoyan, Raga, & Xuemei, 2022) (Xu, Meng, Qiu, Yu, & Wu, 2019). Metode *Stratified Sampling* diterapkan dalam melakukan pembagian sample dataset, hal ini karena dalam *stratified sampling* pembagian sample dilakukan dengan mempertahankan proporsi kategori (kelas) yang ada di dalam dataset, sehingga baik untuk diterapkan pada dataset dengan distribusi kelas yang tidak seimbang.

Tabel 3. 2. Perbandingan Kelas Dataset

Kelas	Positif	Negatif
Perbandingan	36,09%	63,91%
Keseluruhan Data	16.212	28.705
Data Training	13.005	22.928
Data Testing	3.207	5.777

Hasil pembagian sample dataset dengan *stratified sampling* menunjukan jumlah data pada masing-masing kelas dilakukan dengan tetap mempertahankan proporsi awal. Seperti yang disajikan pada Tabel 3.2. keseluruhan data terdiri atas 2 (dua) kelas dengan perbandingan 36,09% dan 63,91%, setelah dilakukan split data untuk *training* dan *testing* jumlah data pada masing-masing kelasnya terbukti tetap memiliki nilai perbandingan yang sama baik untuk data *training* maupun data *testing*. Data tersebut digunakan pada setiap model dalam penelitian ini, sehingga skenario pengujian dilakukan dengan kondisi data yang sama.

6. *Feature Extraction*

Fitur ekstraksi yang digunakan dalam penelitian ini adalah *word embedding*. Adapun *word embedding* yang digunakan pada dataset dalam penelitian ini adalah Word2Vec, FastText, dan GloVe. Penerapan beberapa jenis *word embedding* ini dilakukan dengan tujuan memperoleh representasi kata yang lebih banyak dan variatif (Alghifari, Edi, & Firmansyah, 2022) (Zhao, Liu, Yao, & Yang, 2021) sehingga diharapkan dapat menghasilkan performa model yang lebih baik.

7. *Oversampling* Data

Dikarenakan dataset yang digunakan memiliki kelas yang tidak seimbang, maka diperlukan tahapan penanganan data. Adapun teknik yang digunakan penangan data yang digunakan dalam penelitian ini adalah *oversampling*. Metode *oversampling* dengan SMOTE dipilih karena SMOTE dinilai popular dalam menangani ketidakseimbangan kelas, menghasilkan *instance minoritas* baru dengan menggabungkan *instance* yang sudah ada dan menggunakan interpolasi *linier* untuk menciptakan contoh baru, yang dapat memperluas wawasan kelas minoritas (Chawla, Bowyer, Hall, & Kagelmeyer, 2002).

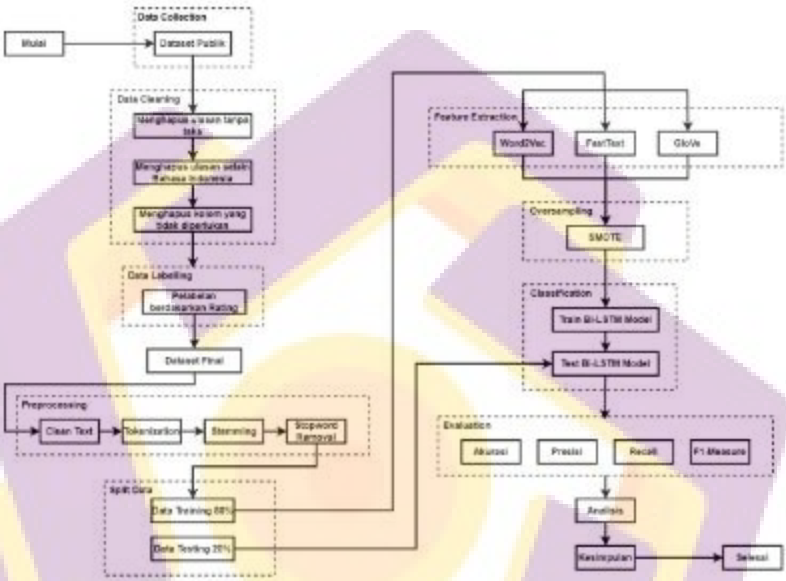
8. *Classification*

Tahapan klasifikasi dalam penelitian ini merupakan tahapan analisis sentimen. Seluruh data akan diklasifikasikan data dengan menggunakan metode BiLSTM. BiLSTM dinilai mampu menangkap informasi dari dua arah sehingga diharapkan mampu meningkatkan model analisis sentimen mengenai permasalahan konvergensi, hal ini bertujuan untuk menghasilkan model dengan performa yang lebih baik (Alsharef, et al., 2022) (Gondhi, et al., 2022) (Riza & Charibaldi, 2021).

9. *Evaluation*

Setelah diperoleh hasil klasifikasi (analisis sentimen) maka dilakukan pengujian performa dari penerapan algoritma *word embedding* dengan menggunakan *confusion matrix* (akurasi, presisi, *recall, f1-score*).

Output tahapan evaluasi ini yang akan menjadi acuan dalam menentukan hasil klasifikasi atau analisis sentimen pada penelitian ini.



Gambar 3. 2. Alur Penelitian

Dalam analisis sentimen, proses fitur ekstraksi seperti *word embedding* dilakukan sebelum proses *oversampling* (Putra & Setiawan, 2023) (Al-Azani & El-Alfy, 2017) (Satriaji & Kusumaningrum, 2018) (Xu, et al., 2015). Tahapan *word embedding* mengubah teks menjadi representasi numerik yang menangkap makna dan konteks kata-kata dalam dimensi yang lebih rendah. Proses ini penting karena algoritma *machine learning* tidak dapat langsung memproses teks mentah, melainkan membutuhkan data dalam bentuk numerik (Nugraha, Harani, & Habibi, 2020), begitupun dengan SMOTE. Sebelum *feature extraction*, data ulasan masih

dalam bentuk teks mentah yang tidak dapat langsung digunakan untuk menerapkan SMOTE. Sehingga teks perlu dikonversi terlebih dahulu ke bentuk numerik melalui proses *feature extraction* dimana dalam penelitian ini dilakukan dengan *word embedding*.

Setelah teks diubah menjadi vektor, SMOTE dapat diterapkan untuk menangani ketidakseimbangan kelas dengan menciptakan contoh sintetis dari kelas minoritas. Terlebih jika SMOTE diterapkan sebelum embedding, hasil sintetisnya tidak akan mempertimbangkan makna semantik dari teks, karena SMOTE bekerja pada data numerik dan tidak dirancang untuk mengolah data teks langsung (Chawla, Bowyer, Hall, & Kagelmeyer, 2002). Oleh karena itu, melakukan *embedding* terlebih dahulu memastikan bahwa data yang seimbang oleh SMOTE sudah berada dalam format yang optimal untuk model *machine learning*, menjaga integritas semantik dan efektivitas proses pembelajaran *oversampling* (Putra & Setiawan, 2023) (Al-Azani & El-Alfy, 2017) (Satriaji & Kusumaningrum, 2018) (Xu, et al., 2015).

BAB IV

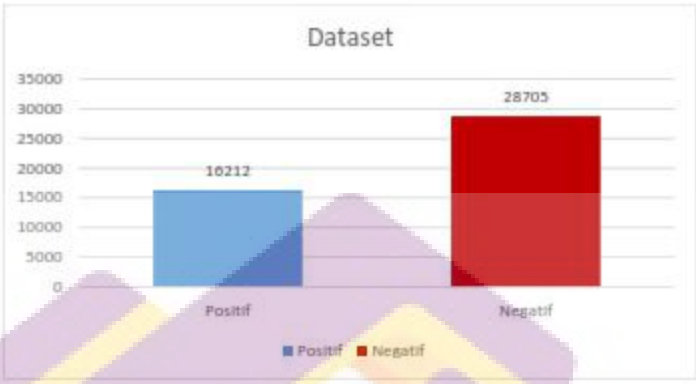
HASIL PENELITIAN DAN PEMBAHASAN

4.1. Data Penelitian

4.1.1. Pengumpulan Data

Penelitian ini menggunakan dataset yang berisi ulasan mengenai suatu aplikasi. Dataset dipilih karena merupakan ulasan aplikasi Gojek yang diungkapkan melalui situr Google Playstore, dimana Gojek merupakan aplikasi *decacorn* buatan Indonesia dan telah diunduh sebanyak lebih dari 100.000.000 kali pada situs Google PlayStore. Serta situs Google Playstore dinilai sebagai sumber terbaik untuk menemukan kumpulan data bersih untuk penelitian analisis sentimen karena berisi ungkapan pengguna yang tidak menyertakan konten komersial atau promosi apapun (Muttaqin & Khasirudin, 2021).

Selain itu dataset yang digunakan terdiri atas 44.917 ulasan pengguna berbahasa Indonesia, hal ini sesuai dengan penelitian sebelumnya, dimana (Alghifari, Edi, & Firmansyah, 2022) yang menggunakan data sejumlah 5.000 ulasan dan (Riza & Charibaldi, 2021) sejumlah 1.304 ulasan menyarankan untuk menggunakan dataset dengan jumlah yang lebih banyak untuk menghindari masalah *overfitting*. Dataset yang digunakan telah memiliki label yang terdiri atas 2 (dua) kelas sentimen, yaitu Positif dan Negatif. Label positif mewakili ulasan pengguna yang memiliki penilaian yang cenderung baik, sedangkan label negatif mewakili ulasan pengguna yang memiliki penilaian yang cenderung buruk terhadap aplikasi Gojek. Kondisi kelas dataset divisualisasikan pada Gambar 4.1.



Gambar 4. 1. Kelas Dataset

Kondisi distribusi data pada Gambar 4.1 merupakan dataset final yang telah melalui tahapan *cleaning/filtering*. Tahapan *cleaning/filtering* seperti memfilter data ulasan yang tidak memiliki teks ulasan, memfilter data ulasan selain Bahasa Indonesia, dan menghapus kolom-kolom yang tidak diperlukan menghasilkan dataset seperti contoh yang disajikan pada Tabel 4.1.

Tabel 4. 1. Contoh Dataset

Content	Label
Ini aplikasinya gimana? Kemarin2 gk ada fitur tarik tunai terus habis update ini muncul lagi fiturnya tpi gk bisa digunakan, kode transaksi nya gk muncul. Saya udah mencoba berulang kali tetep gk muncul kode transaksinya. Tolong dong diperbaiki lagi!!!	Negatif
kode otp gak masuk masuk, udah update terbaru, udah clear cache, tetep gak bisa masuk, udah on-off airplane tetep gak masuk, padahal sinyal full, udah coba kirim ulang berkali kali juga gak masuk, jadi gak bisa top up dengan oneklik bca.	Negatif
Dulu: Banyak bonus. Bisa pesan lintas kota. kurangnya cuma yg super partner, klo menu ga sesuai, ga bisa di cancel walaupun resto dah konfirmasi. Sekarang: "Lebih mahal ongkos kirim dan biaya lain-lain" daripada harga satuan barang/makanan tsb. Ga logis. Dan ga ada diskon walau beli banyak s.d 200rban, klo mau ada diskon harus beli voucher dulu.	Negatif
Driver2nya bs antar dg tepat, ga bingung arah dan tdk salah pesanan, tdk spt pengalaman saya di aplikasi lainnya. User friendly dan lengkap. Go pay jg sgt membantu. Skrg driver2nya sangat responsif saat hujan, terimaksh masukan saya didengarkan. Mode hemat jg sangat membantu saat kantong tipis.	Positif

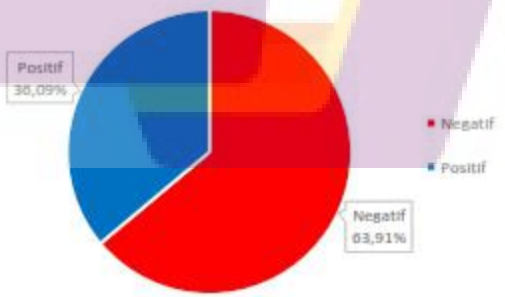
Tabel 4. 1. Contoh Dataset (Lanjutan)

Content	Label
Akan lebih baik jika lokasi tidak disetting otomatis di lokasi sekitar. Boleh jadi, pelanggan ingin mengecek biaya gocar/goride suatu tempat dari jarak jauh. Misal, bandara ke hotel. Setting lokasi sekitar mengakibatkan pengecekan biaya tak dapat dilakukan. Sebab, lokasi yang jauh tak terdeteksi.	Negatif
Performa makin lama makin menurun dan harga yg di tawarkan juga lebih tinggi dibanding aplikasi sebelah coba diperbaiki sistemnya lagi seperti awal jangan kebanyakan upgrade aplikasi makin lemot dan berat keseringan di perbaharui.	Negatif
Aplikasi tokai. Makin lama makin menurun performanya. Biaya layanan aja digedein tpi aplikasi makin anjlok. terus biaya layanan buat apa??? Tiba2 ngecancel sendiri pas udah lama nunggu. Dimuat ulang berulang kali juga gak bisa. Smpe restart hp dan coba pesen lagi yaa tetep gak bisaaaa. Terus maunyaa apaaa anjerrr Otw uninstall	Negatif
Pemesanan go-food kurang memuaskan,sistem kupon terpakai secara otomatis menyulitkan. Saldo dan pesanan minimum sudah mencukupi namun kupon tidak dapat terpakai. Lokasi pengiriman makanan tidak sesuai GPS, ketika salah pilih tidak dapat di cancel, dikenakan charge tambahan sama driver karena tujuan pengantaran berbeda. Kecewa dengan kebijakan sekarang.	Negatif
Untuk gofood Kenapa driver yang dipilih itu bukan driver yg dekat dengan resto, malah yg dipilih bisa yg jaraknya dri resto itu sangat jauh sekali. Untuk go ride juga demikian, padahal keliatan driver banyak di dekat lokasi, tapi yg terpilih driver yg jaraknya jauh sekali dari titik penjemputan. Akhirnya kami disuruh batalkan pesanan untuk cari driver yg lebih dekat. . Mohon diperbaiki. .	Negatif
Semakin Update versi semakin memperbanyak Peraturan.. Bagus sih..!! Tapi Untuk Aplikator .. Saya kasihan dengan Mitra kalian yang Rebahan di jalan arau bahkan Meluangkan waktu nya sampai Larut Malam hanya untuk Berkontribusi dengan aplikator.. Kalian pantau lah mereka..!! Jangan hanya memperbanyak biaya ini dan itu yang membuat semakin banyak Para Consumen enggan untuk menggunakan aplikasi ini, Yang berdampak Ke Mitra kalian secara Real time Yyang onbit hingga larut malam	Positif
Sistem upgrade plus yg sangat buruk. Cuma bisa topup, tapi dipakai untuk transfer dan sejenisnya tidak bisa. Ktp dan foto sudah benar sesuai persyaratan, tapi berkali kali ditolak. Sangat jauh dibandingkan dengan aplikasi DANA atau OVO. Aplikasi gojek hanya cocok untuk order ojek dan service saja	Negatif

4.1.2. Ragam Sentimen Ulasan Gojek pada Playstore

Berdasarkan dataset yang diperoleh melalui situs penyedia dataset yaitu Kaggle.com mengenai ulasan pengguna aplikasi Gojek yang diungkapkan pengguna pada situs Google Playstore, diperoleh data ulasan sebanyak 44.917 ulasan dengan ragam sentimen positif dan negatif.

Ragam sentimen terdiri atas 16.212 ulasan dengan sentimen positif atau sebesar 36,09% dan 28.705 ulasan dengan senitmen negatif atau sebesar 63,91%. Visualisasi ragam sentimen tersebut disajikan pada Gambar 4.2, dimana sentimen positif direpresentasikan menggunakan warna biru dan sentimen negatif direpresentasikan dengan warna merah. Berdasarkan visualisasi pada Gambar 4.2 dapat disimpulkan bahwa ulasan negatif yang diberikan oleh pengguna aplikasi Gojek melalui situs penyedia aplikasi yakni Google Playstore berjumlah lebih banyak dibandingkan ulasan beragam sentimen positif, yaitu dengan selisih sebesar 27,82%. Nilai selisih tersebut diperoleh dari perbedaan banyaknya ragam sentimen masing-masing kelas, dalam hal ini yaitu banyaknya data dengan kelas negatif dikurangi dengan banyaknya data dengan kelas positif. Dengan adanya nilai selisih pada ragam sentimen, kelas data dalam penelitian ini dinilai memiliki kelas yang tidak seimbang (*imbalanced dataset*). Kondisi tersebut akan ditindaklanjuti dengan teknik *oversampling*, sehingga dalam penelitian ini akan diterapkan data dengan 2 (dua) kondisi berbeda yaitu data original dengan kelas tidak seimbang (*imbalanced data*) dan data dengan kelas seimbang (setelah *oversampling*).



Gambar 4. 2. Ragam Sentimen

4.2. Mekanisme/Skenario Training Data

Proses eksperimen pada penelitian ini diawali dengan persiapan data, Adapun persiapan data dilakukan dengan menerapkan beberapa tahapan teknik *preprocessing data*, seperti *cleaning data*, *tokenization*, *stemming*, dan *stopword removal*.

Setelah melalui *preprocessing*, data divektorisasikan dengan menggunakan 3 (tiga) jenis *word embedding* yaitu Word2Vec, FastText, dan GloVe. Kemudian hasil model vektorisasi yang dihasilkan pada tahapan tersebut akan diklasifikasikan dengan menggunakan model Bi-LSTM, model Bi-LSTM dinilai mampu menangkap informasi dua arah sehingga mampu meningkatkan model dalam mencapai konvergensi. (Alsharef, et al., 2022) (Gondhi, et al., 2022) (Riza & Charibaldi, 2021) dan menghasilkan performa yang baik dalam melakukan klasifikasi, namun menurut (Alghifari, Edi, & Firmansyah, 2022) Bi-LSTM membutuhkan waktu dan komputasi yang cukup tinggi. Sehingga dilakukan perbandingan terhadap penerapan 3 (tiga) jenis *word embedding* untuk mengetahui jenis word embedding yang mampu menghasilkan representasi kata dan performa yang lebih baik.

Penelitian ini memanfaatkan *environment* dari Google Collaboratory dalam melakukan eksperimen. Serta menggunakan *framework* Keras, tensor Flow, Sckit Learn, dan NLTK. Selanjutnya untuk menetukan nilai *epoch* dalam melakukan training dilakukan berdasarkan eksperimen dengan menggunakan nilai *epoch* mulai dari 30, 50, dan 100. Performa model saat *epoch* 50 memiliki nilai yang cukup stabil dan tidak mengalami peningkatan lagi, sehingga *epoch* 50 digunakan pada seluruh

skenario. Selanjutnya penelitian ini menerapkan teknik monitoring *early stopping* dalam pengelolaan pelatihan model karena mampu mencegah terjadinya *overfitting*.

Setiap model *word embedding* menerapkan beberapa parameter dalam membangun model, seperti arsitektur, *min_count*, dan metode evaluasi. Arsitektur CBOW (*Continuous Bag of Words*) dipilih karena kata target yang diprediksi oleh CBOW dilakukan berdasarkan kata atau konteks di sekitarnya. Sehingga CBOW, tidak mempermasalahkan letak sebuah kata di antara kata-kata di sekitarnya. pendekatan ini memprediksi kata sasaran berdasarkan konteks (Afidah, Handayani, & Pratiwi, 2021). CBOW juga dinilai mampu menghasilkan akurasi yang lebih besar dengan waktu pelatihan yang lebih sedikit (Rong, 2014). Nilai *min count* yang digunakan adalah 10, hal ini berarti agar suatu kata dapat dimasukkan kedalam kamus yang digunakan untuk pelatihan model harus memiliki jumlah minimum kemunculan sebanyak 10.

Pada model Word2Vec dan FastText terdapat 3 (tiga) parameter yang menjadi pembeda pada setiap skenario yang dilakukan, yaitu kondisi data pada proses *training*, jumlah *window*, dan *embedding size* (dimensi) yang digunakan. Sedangkan model GloVe hanya terdiri atas 2 (dua) parameter utama yang menjadi pembeda, yaitu kondisi data dan *embedding size* (dimensi).

Data yang digunakan pada proses *training* terdiri atas 2 (dua) kondisi data, yaitu data tidak seimbang (*original*) dan data yang telah diterapkan *oversampling* dengan SMOTE (seimbang). Parameter selanjutnya adalah mengenai banyaknya *window* yang digunakan, pada penelitian ini menggunakan 3 (tiga) nilai *window* berbeda, yaitu 3, 5, dan 7. Sedangkan untuk *embedding size* (dimensi)

menggunakan nilai sebesar 100 dan 300. Berdasarkan parameter yang digunakan dalam penelitian ini diperoleh variasi sebanyak 28 skenario seperti pada Tabel 4.2, dimana untuk setiap skenario dilakukan pengujian sebanyak 3 (tiga) kali untuk memperkuat analisa performa model. Setiap skenario dilakukan pada file *notebook* yang berbeda sehingga eksperimen dapat dilakukan secara efektif.

Tabel 4. 2. Variasi Skenario

Word Embedding	Data	Window			Dimensi	
		3	5	7	100	300
Word2Vec	Tidak Seimbang	√	-	-	√	-
		-	√	-	√	-
		-	-	√	√	-
		√	-	-	-	√
		-	√	-	-	√
		-	-	√	-	√
	Seimbang (SMOTE)	√	-	-	√	-
		-	√	-	√	-
		-	-	√	√	-
		√	-	-	-	√
		-	√	-	-	√
		-	-	√	-	√
FastText	Tidak Seimbang	√	-	-	√	-
		-	√	-	√	-
		-	-	√	√	-
		√	-	-	-	√
		-	√	-	-	√
		-	-	√	-	√
	Seimbang (SMOTE)	√	-	-	√	-
		-	√	-	√	-
		-	-	√	√	-
		√	-	-	-	√
		-	√	-	-	√
		-	-	√	-	√
GloVe	Tidak Seimbang	-	-	-	√	-
		-	-	-	√	-
	Seimbang (SMOTE)	-	-	-	-	√
		-	-	-	-	√

Variasi skenario pada Tabel 4.2 akan diterapkan untuk tahapan pengujian model analisis sentimen, dimana hasil pengujian tersebut akan didisajikan pada Tabel 4.16 dan digunakan untuk melakukan analisa terhadap hasil klasifikasi model.

4.3. Preprocessing

Tahapan *preprocessing* dalam penelitian ini dilakukan untuk membersihkan dataset dari data-data yang tidak diperlukan untuk pemrosesan selanjutnya. Proses pembersihan data dilakukan dengan menghapus data ulasan yang tidak memiliki teks ulasan, menghapus data ulasan yang berbahasa selain Bahasa Indonesia, dan menghapus kolom yang tidak diperlukan. *Preprocessing* yang digunakan dalam penelitian ini antara lain *Clean Text*, Tokenization, *Stemming* dengan Sastrawi, dan *Stopword Removal*. Hasil preprocessing pada masing-masing tahapan disajikan pada Tabel 4.3.

4.3.1. Clean Text

Pada tahap ini, data ulasan dibersihkan dari data yang tidak diperlukan seperti tanda baca, angka, simbol, *emoticon*, dan semacamnya. Selain itu, *casefolding* juga diterapkan pada tahapan ini sehingga data ulasan diubah menjadi lowercase.

4.3.2. Tokenization

Proses tokenisasi ini membagi teks menjadi token-token, yaitu kata-kata atau frasa secraa terpisah, dan menghilangkan tanda baca. Hasil tokenisasi dapat

digunakan sebagai langkah awal dalam analisis teks atau pemrosesan bahasa alami lebih lanjut seperti perhitungan vektor setiap kata.

4.3.3. Stemming

Proses *stemming* bertujuan untuk mengembalikan kata-kata ke bentuk dasarnya atau kata dasar. Seperti "memiliki" menjadi "milik", "diperbaiki" menjadi "baik", dan sebagainya. *Stemming* membantu mengurangi variasi kata dan dapat meningkatkan konsistensi dalam analisis teks. Tahapan *stemming* ini menerapkan aturan morfologi Bahasa Indonesia, yaitu *stemming* dengan metode Sastrawi. Sastrawi-stemmer mampu menunjukkan hasil tertinggi dalam mengembalikkan kata-kata berimbuhan kedalam bentuk kata dasar yang beracuan pada KBBI saat dibandingkan dengan Nazief-Andriani dan Arifin-Setiono (Mustikasari, Widaningrum, Arifin, & Putri, 2021).

4.3.4. Stopword Removal

Berdasarkan kondisi data dalam penelitian ini menggunakan data ulasan berbahasa Indonesia, maka *stopword* "Indonesian" diterapkan pada tahapan *preprocessing*. *Stopword* seperti "yang", "untuk", "dengan", "ke", "mana", "saja", "jika", "kita", "tidak", "atau", "dan", "adalah", "suatu", "jadi", "saya", "itu", "dan", "saya", "dong" telah dihapus dari teks ulasan tersebut. Hasil *stopword* dapat digunakan untuk analisis lebih lanjut tanpa adanya kata-kata yang dianggap umum atau kurang informatif. Meskipun tidak terlalu signifikan, penerapan *stopword removal* dinilai mampu menghasilkan peningkatan akurasi dalam klasifikasi (Hidayatullah, 2016)

Tabel 4. 3. *Preprocessing* Data

	Setelah Preprocessing
Original	Ini aplikasinya gimana? Kemarin2 gk ada fitur tarik tunai terus habis update ini muncul lagi fiturnya tpi gk bisa digunakan, kode transaksi nya gk muncul. Saya udah mencoba berulang kali tetep gk muncul kode transaksinya. Tolong dong diperbaiki lagi!!!
Clean Text	ini aplikasinya gimana kemarin gak ada fitur tarik tunai terus habis update ini muncul lagi fiturnya tapi gak bisa digunakan kode transaksi nya gak muncul Saya udah mencoba berulang kali tetep gak muncul kode transaksinya tolong dong diperbaiki lagi
Tokenization	['ini', 'aplikasinya', 'gimana', 'kemarin', 'gak', 'ada', 'fitur', 'tarik', 'tunai', 'terus', 'habis', 'update', 'ini', 'muncul', 'lagi', 'fiturnya', 'tapi', 'gak', 'bisa', 'digunakan', 'kode', 'transaksi', 'nya', 'gak', 'muncul', 'saya', 'udah', 'mencoba', 'berulang', 'kali', 'tetep', 'gak', 'muncul', 'kode', 'transaksinya', 'tolong', 'dong', 'diperbaiki', 'lagi']
Stemming	['ini', **'aplikasi'**, 'gimana', 'kemarin', 'gak', 'ada', 'fitur', 'tarik', 'tunai', 'terus', 'habis', 'update', 'ini', 'muncul', 'lagi', **'fitur'**, 'tapi', 'gak', 'bisa', **'guna'**, 'kode', 'transaksi', 'nya', 'gak', 'muncul', 'saya', 'udah', **'coba'**, **'ulang'**, 'kali', 'tetep', 'gak', 'muncul', 'kode', **'transaksi'**, 'tolong', 'dong', **'baik'**, 'lagi']
Stopword Removal	['aplikasi', 'kemarin', 'fitur', 'tarik', 'tunai', 'habis', 'update', 'muncul', 'fitur', 'bisa', 'guna', 'kode', 'transaksi', 'muncul', 'coba', 'ulang', 'tetep', 'muncul', 'kode', 'transaksi', 'tolong', 'baik']

Data hasil *preprocessing* tersebut nantinya akan digunakan untuk membangun corpus hingga membuat *vocabulary* yang akan dimanfaatkan pada proses fitur ekstraksi dan *training data*. Berikut merupakan contoh tahapan membangun corpus dari 3 (tiga) buah ulasan pada Tabel 4.4.

Tabel 4. 4. Pembuatan Corpus

Tahapan	**Hasil**
Kalimat Ulasan	"Saya udah mencoba berulang kali tetep gk muncul kode transaksinya" "Performa makin lama makin menurun dan harga yg di tawarkan juga lebih tinggi" "Mode hemat jg sangat membantu saat kantong tipis"
Setelah Tokenisasi	['saya', 'udah', 'mencoba', 'berulang', 'kali', 'tetep', 'gk', 'muncul', 'kode', 'transaksinya'], ['performa', 'makin', 'lama', 'makin', 'menurun', 'dan', 'harga', 'yg', 'di', 'tawarkan', 'juga', 'lebih', 'tinggi'], ['mode', 'hemat', 'jg', 'sangat', 'membantu', 'saat', 'kantong', 'tipis']
Corpus yang Dihasilkan	['saya', 'udah', 'mencoba', 'berulang', 'kali', 'tetep', 'gk', 'muncul', 'kode', 'transaksinya', 'performa', 'makin', 'lama', 'menurun', 'dan', 'harga', 'yg', 'di', 'tawarkan', 'juga', 'lebih', 'tinggi', 'mode', 'hemat', 'jg', 'sangat', 'membantu', 'saat', 'kantong', 'tipis']

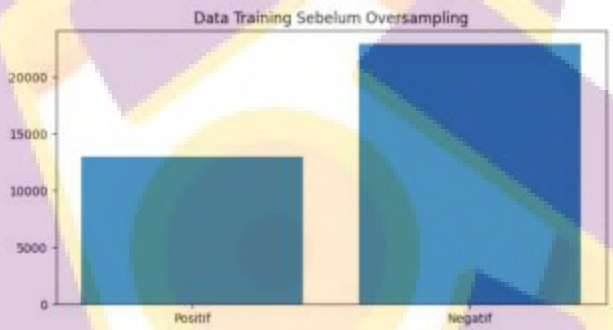
4.4. Penanganan *Imbalanced* Data

Penanganan data tidak seimbang diterapkan pada dataset yang digunakan untuk proses *training*, yaitu sebanyak 35.928 data yang merupakan 80% dari keseluruhan dataset. Sehingga data yang diproses dalam tahapan ini terdiri atas 13.005 ulasan positif dan 22.928 ulasan negatif, sehingga data memiliki kondisi yang tidak seimbang. Untuk mengetahui seberapa besar pengaruh kondisi data terhadap performa model maka diperlukan penanganan terhadap data yang tidak seimbang.

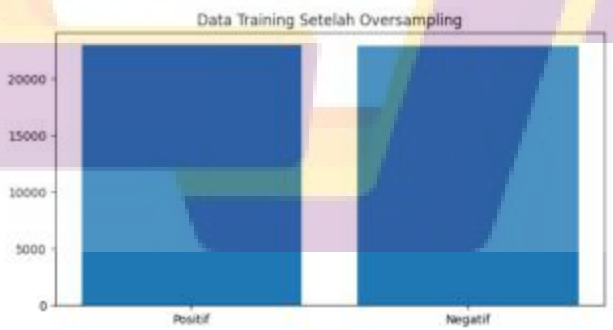
Metode penanganan yang diterapkan pada data dalam penelitian ini merupakan salah satu metode *oversampling*, yaitu SMOTE (*Synthetic Minority Oversampling Technique*). SMOTE cukup populer dalam mengatasi masalah ketidakseimbangan. *Instance minoritas* baru dihasilkan oleh SMOTE dengan menggabungkan *instance minoritas* yang sudah ada sebelumnya. Contoh minoritas baru dibuat untuk kelas minoritas melalui penggunaan interpolasi linier sehingga dinilai mampu memperluas wawasan kelas minoritas (Chawla, Bowyer, Hall, & Kagelmeyer, 2002).

Distribusi dataset *training* yang semula terdiri atas positif sebanyak 13.005 dan negatif sebanyak 22.928, berubah menjadi 22.928 (negatif), 22.928 (positif) setelah diterapkannya *oversampling* menggunakan SMOTE. Perbedaan perbandingan ragam data sebelum dan setelah *oversampling* dapat dilihat pada Gambar 4.3 dan Gambar 4.4. Pelatihan virtual mengenai penanganan data dalam penelitian dilakukan dengan mengatur nilai parameter k dengan memanfaatkan pendekatan berbasis *grid search* mulai dari k=1,3,5, hal ini dinilai mampu

memberikan parameter k yang optimal karena *grid search* menerapkan berbagai nilai k secara otomatis sehingga dihasilkan parameter optimal berdasarkan karakteristik dan ukuran data yang sedang diproses. Nilai k=3 merupakan parameter yang tepat dalam eksperimen ini sehingga dapat menghasilkan data sintetik yang lebih representative dan mengurangi resiko *overfitting* maupun *underfitting*. Setelah proses oversampling, data dibangun kembali, dan data yang dimodifikasi dapat diterapkan pada model klasifikasi analisis sentimen.



Gambar 4. 3. Data Training Sebelum SMOTE



Gambar 4. 4. Data Training Setelah SMOTE

4.5. Model Word2Vec

Word2vec mengandalkan informasi lokal berdasarkan semantik yang dipelajari dari suatu kata tertentu dimana kata tersebut dipengaruhi oleh kata-kata di sekitarnya. Wod2vec menunjukkan kemampuan untuk mempelajari pola linguistik sebagai hubungan linier antara vektor kata (Dharma, Gaol, Warnars, & Soewito, 2022). Terdapat 2 (dua) model arsitektur yang dapat digunakan di Word2Vec yaitu *Continuous Bag-of-Word* (CBOW) dan Skip-Gram. Dalam penelitian ini diterapkan arsitektur CBOW, yaitu dalam memprediksi kata target dilakukan berdasarkan kata konteks disekitarnya (Nurdin, Aji, Bustamin, & Abidin, 2020).

Proses pelatihan model Word2Vec terdiri atas beberapa tahapan sebagai berikut. Tahapan *vocabulary* dan *one-hot encoding* ini dilakukan berdasarkan kumpulan kata yang telah dihasilkan pada corpus (Tabel 4.5). Daftar kata unik (*vocabulary*) dalam corpus tersebut selanjutnya diubah menjadi vektor *one-hot encoding*.

Selanjutnya dalam menentukan kata target dan kata konteks pada setiap kata dalam kalimat dilakukan dengan arsitektur CBOW dan *window size*. Berikut contoh pasangan kata-kata yang digunakan sebagai input dan output dalam pelatihan model Word2Vec dengan arsitektur CBOW dengan *window size*=3.

Gambar 4. 5. Pembentukan Konteks dan Target CBOW

Berdasarkan pelatihan yang digambarkan pada Gambar 4.5 diperoleh daftar pasangan kata-kata yang berperan sebagai *input* (konteks) dan *output* (target) dalam kalimat ulasan. Adapun daftar pasangan kata konteks dan target disajikan pada Tabel 4.5.

Tabel 4. 5. Hasil Pasangan Konteks dan Target CBOW

Pelatihan Word2Vec dengan arsitektur CBOW	
Input	**Output**
Context: [udah, mencoba, berulang]	Target: saya
Context: [saya, mencoba, berulang, kali]	Target: udah
Context: [saya, udah, berulang, kali, tetep]	Target: mencoba
Context: [saya, udah, mencoba, kali, tetep, gk]	Target: berulang
Context: [udah, mencoba, berulang, tetep, gk, muncul]	Target: kali
Context: [mencoba, berulang, kali, gk, muncul, kode]	Target: tetep
Context: [berulang, kali, tetep, muncul, kode, transaksinya]	Target: gk
Context: [kali, tetep, gk, kode, transaksinya]	Target: muncul
Context: [tetep, gk, muncul, transaksinya]	Target: kode
Context: [gk, muncul, kode]	Target: transaksinya

Setelah diperoleh konteks dan target kata selanjutnya dilakukan pembobotan kata dengan inisialisasi bobot secara acak, menghitung skor dan probabilitas menggunakan *softmax*, menggunakan algoritma pembaruan bobot (seperti *gradient descent*), dan melatih model dengan *forward* dan *backward propagation* untuk memperbarui bobot berdasarkan *error*.

4.5.1. Pembuatan Model Word2Vec

Proses pembuatan model Word2Vec dalam penelitian ini dilakukan dengan memanfaatkan *libary* yang disediakan oleh Phyton, yakni Gensim. Word2Vec memiliki parameter yang digunakan untuk membangun model, beberapa parameter seperti arsitektur, metode evaluasi, dan dimensi dinilai cukup berpengaruh terhadap kinerja *deep learning* (Af'idah, Handayani, & Pratiwi, 2021). Sehingga dalam penelitian ini model Word2Vec yang digunakan dibangun dengan menerapkan parameter arsitektur, *minimum count*, hingga metode evaluasi yang bernilai atau berjenis sama, dan parameter *window* serta *embedding size* (dimensi) dengan nilai atau jenis yang berbeda seperti pada Tabel 4.6.

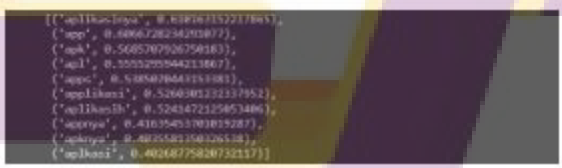
Tabel 4. 6. Model Word2Vec

Model Word2Vec	
Parameter	**Jenis/Nilai yang Digunakan**
Arsitektur	CBOW (sg=0)
Size/Dimensi	100, 300
Window	3, 5, 7
Min Count	10
Metode Evaluasi	*Hierarchical Softmax*

Jenis arsitektur CBOW dan metode evaluasi *hierarchical softmax* dipilih karena kedua parameter tersebut mampu menghasilkan performa yang unggul

dalam melakukan klasifikasi sentimen pada penelitian terdahulu (Af'idah, Handayani, & Pratiwi, 2021). *Minimum count* diterapkan untuk menentukan berapa banyak suatu kata harus muncul saat proses pelatihan model agar kata tersebut dimasukkan ke dalam kamus, nilai *minimum count* yang diterapkan adalah 10. *Embedding size* merupakan ukuran vektor yang dihasilkan dari suatu kata yang akan digunakan untuk mewakili setiap kata tersebut dalam teks, nilai *embedding size* yang diterapkan adalah 100 dan 300. Kemudian parameter *window* diterapkan untuk menentukan jumlah kata yang perlu diperhatikan yang terletak sebelum dan setelah kata target, nilai *window* yang digunakan adalah 3, 5, dan 7.

Model Word2Vec yang dibangun mampu mengembalikan suatu kata beserta skor kemiripan kata tersebut dengan kata dasar. Pada Gambar 4.6 menunjukkan daftar tupel dari kata dasar "aplikasi" beserta skor kemiripannya dengan kata "aplikasi". Kata "aplikasinya" yang merupakan tupel dari kata dasar memiliki skor kemiripan sebesar 0,610163152217865. Semakin tinggi skor kemiripannya, semakin mirip kata tersebut dengan "aplikasi".



Gambar 4. 6. Hasil Kemiripan Kata "Aplikasi" pada Model Word2Vec

4.5.2. Training dan Testing Word2Vec

Pada tahap ini, prosedur *training* dan *testing* dilakukan dengan pelatihan model menggunakan *word embedding* Word2Vec dalam melakukan ekstraksi fitur, dimana setiap skenario dijalankan sebanyak 3 (tiga) kali untuk memperkuat hasil

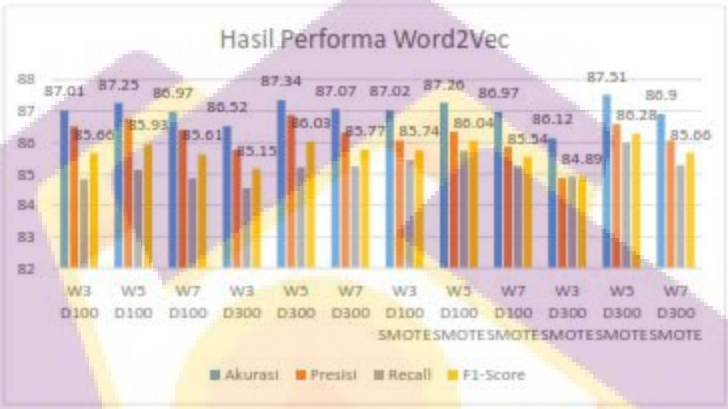
analisa. Data ulasan yang diekstraksi kemudian diklasifikasikan menggunakan Bi-LSTM. Penerapan parameter-parameter model Word2Vec beserta keadaan dataset seimbang dan tidak seimbang menghasilkan kinerja klasifikasi analisis sentimen seperti yang disajikan pada Tabel 4.5, hasil yang disajikan merupakan rata-rata performa untuk setiap skenario. Pendekatan *EarlyStopping* digunakan sebagai langkah pelatihan prosedur monitoring dalam penelitian ini.

Pengukuran kinerja pada model dengan kondisi dataset dengan jumlah data yang seimbang pada tiap kelasnya dinilai berdasarkan perolehan nilai akurasi, kinerja terbaik pada model Word2Vec dicapai saat menerapkan parameter *window*=5 dan *embedding size*=300 dengan rata-rata akurasi sebesar 87,51%. Sementara pada kondisi tidak seimbang pengukuran kinerja dilakukan berdasarkan perolehan nilai *f1-score*, dimana pada Tabel 4.7 kinerja terbaik diperoleh pada model Word2Vec dengan parameter nilai *window*=5 dan *embedding size*=300 dengan perolehan nilai *f1-score* sebesar 86,03%.

Tabel 4. 7. Hasil Performa Word2Vec

Word Embedding	Data	Window			Dimensi		Rata-rata Performa			
		3	5	7	100	300	Akurasi	Presisi	Recall	F1-Score
Word2Vec	Tidak Seimbang (Original)	√	-	-	√	-	87,01	86,50	84,83	85,66
		-	√	-	√	-	87,25	86,75	85,13	85,93
		-	-	√	√	-	86,97	86,39	84,86	85,61
		√	-	-	-	√	86,52	85,75	84,55	85,15
		-	√	-	-	√	87,34	86,87	85,20	86,03
		-	-	√	-	√	87,07	86,31	85,23	85,77
	Seimbang (SMOTE)	√	-	-	√	-	87,02	86,05	85,43	85,74
		-	√	-	√	-	87,26	86,34	85,73	86,04
		-	-	√	√	-	86,97	85,86	85,22	85,54
		√	-	-	-	√	86,12	84,87	84,91	84,89
		-	√	-	-	√	87,51	86,58	85,99	86,28
		-	-	√	-	√	86,90	86,05	85,27	85,66

Perbandingan perolehan hasil performa penerapan *word embedding* Word2Vec memiliki selisih perbedaan yang tidak terlalu siginifikan. Hal ini dapat dilihat pada grafik hasil pengujian dengan variasi parameter Word2Vec yang divisualisasikan pada Gambar 4.7.



Gambar 4. 7. Perbandingan Kinerja Variasi Parameter Word2Vec - Bi-LSTM

Word2Vec mampu diterapkan untuk melatih data pada corpus teks yang besar dan menghasilkan representasi kata yang berkualitas tinggi dengan menangkap hubungan semantik. Namun karena Word2Vec hanya mempelajari representasi untuk kata-kata secara penuh, sehingga tidak efektif untuk kata-kata yang jarang muncul atau kata-kata yang tidak terdapat pada corpus pelatihan.

4.6. Model FastText

Pendekatan model FastText menyelidiki representasi kata dengan memasukkan informasi sub kata. Setiap kata dalam *vocabulary* diwakili oleh urutan karakter *n-gram* yang memungkinkan pemahaman kata-kata yang lebih pendek dan

pemahaman sufiks dan awalan kata. Ilustrasi pemecahan kata dengan *n-gram* disajikan pada Tabel 4.8, dimana nilai n-gram yang digunakan adalah n=3. Setiap karakter *n-gram* memiliki representasi vektor, dan kata-kata direpresentasikan sebagai total representasi vektor tersebut. Seperti halnya vektor untuk kata 'saya' merupakan total representasi dari vector '<sa', 'say', 'aya', 'ya>'.

Tabel 4. 8. Pemecahan Kata dengan N-gram

Pembuatan N-gram pada FastText	
Kata	**n-gram (n=3)**
saya	'<sa', 'say', 'aya', 'ya>'
udah	'<ud', 'uda', 'dah', 'ah>'
mencoba	'<me', 'men', 'enc', 'nco', 'cob', 'oba', 'ba>'
berulang	'<be', 'ber', 'eru', 'rul', 'ula', 'lan', 'ang', 'ng>'
kali	'<ka', 'kal', 'ali', 'li>'
tetep	'<te', 'tet', 'ete', 'tep', ep>'
gk	'<gk', 'gk>'
muncul	'<mu', 'mun', 'unc', 'ncu'. 'cul', 'ul>'
kode	'<ko', 'kod', 'ode', 'de>'
transaksinya	'<tr', 'tra', 'ran', ans', 'nsa', 'sak', 'aks', 'ksi', 'sin', 'iny', 'nya', ya>'

Meskipun dalam pelatihan model FastText memiliki arsitektur yang mirip dengan arsitektur CBOW pada Word2Vec, namun dalam merepresentasikan kata kedalam bentuk vektor dilakukan dengan memperhatikan setiap karakter *n-gram* dan morfologi kata. Seperti pada tahapan *one-hote encoding* setiap *n-gram* diwakili oleh vektor *one-hot*, hal ini menghasilkan vektor yang untuk setiap kata memiliki. Sehingga FastText mampu melatih model dengan cepat pada kumpulan data besar dan memberikan representasi untuk kata-kata yang tidak ada dalam data pelatihan. Dalam menangani kata yang tidak muncul selama pelatihan model, FastText menentukan penyematan vektor dengan memecahnya menjadi *n-gram* seperti pada Tabel 4.8.

4.6.1. Pembuatan Model FastText

Model FastText dalam penelitian ini menerapkan pelatihan jenis *self-train*, dimana pada proses pelatihannya menggunakan dataset yang digunakan dalam penelitian sehingga model ini menggunakan corpus dengan Bahasa Indonesia yang digunakan sehari-hari, berbeda dengan model *pre-trained* yang menggunakan corpus Wikipedia. Penerapan model *self-train* dinilai sesuai dengan gaya Bahasa yang digunakan pengguna dalam memberikan ulasan, yakni Bahasa Indonesia yang tidak terlalu baku.

Proses pembuatan model FastText dilakukan dengan memanfaatkan *libary* yang disediakan oleh Phyton, yakni Gensim. Parameter lain yang digunakan dalam pembuatan model FastText adalah arsitektur, *embedding size*, *window*, *minimum count*, dan metode evaluasi. Nilai dan jenis parameter yang digunakan dalam penelitian disajikan pada Tabel 4.9.

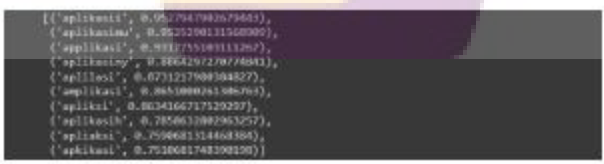
Tabel 4. 9. Model FastText

Model FastText	
Parameter	**Jenis/Nilai yang Digunakan**
N-gram	3
Size/Dimensi	100, 300
Window	3, 5, 7
Min Count	10

Beberapa parameter dalam FastText memiliki sedikit perbedaan dengan Word2Vec, meskipun menerapkan arsitektur yang serupa seperti CBOW, dalam FastText arsitektur ini mengalami pengembangan dengan mempertimbangkan *subword information* (morfologi kata) dalam pembentukan vektor kata, adapun

dalam penelitian ini nila *n-gram* yang digunakan adalah 3. *Embedding size* untuk menentukan jumlah hasil vektorisasi suatu kata kedalam bentuk angka, *embedding size* yang digunakan adalah 100 dan 300, sehingga sebuah kata akan divektorisasi atau diubah menjadi sebanyak 100 atau 300 angka. *Window* adalah banyaknya kata yang akan dicek sebelum dan setelah kata target, nilai *window* yang digunakan adalah 3, 5, dan 7, hal ini berarti setiap kata akan dicek sebanyak n kekiri dan kekanan dari letak kata yang ada. *Minimum count* untuk menentukan minimal jumlah kata yang muncul agar divektorisasi dan dimasukkan kedalam *vocab*, dalam hal ini nilai yang digunakan adalah 10. *Hierarchical softmax* adalah suatu metode yang digunakan dalam FastText untuk mengurangi kompleksitas perhitungan saat melakukan prediksi. Metode ini mengubah distribusi probabilitas dari seluruh kelas menjadi hierarki, sehingga dapat mempercepat proses prediksi.

Model FastText yang dibangun mampu mengembalikan suatu kata beserta skor kemiripan kata tersebut dengan kata dasar. Pada Gambar 4.8 menunjukkan daftar tupel dari kata dasar "aplikasi" beserta skor kemiripannya dengan kata "aplikasi". Kata "aplikasii" yang merupakan tupel dari kata dasar memiliki skor kemiripan sebesar 0,952794902679443. Semakin tinggi skor kemiripannya, semakin mirip kata tersebut dengan "aplikasi".



Gambar 4. 8. Hasil Kemiripan Kata "Aplikasi" pada Model FastText

4.6.2. Training dan Testing FastText

Proses *training* dan *testing* yang dilakukan pada tahapan ini diawalai dengan pelatihan model dengan penerapan *word embedding* FastText sebagai ekstraksi fitur, kemudian data hasil ekstraksi akan diklasifikasikan dengan Bi-LSTM. Kombinasi penerapan parameter model FastText dan kondisi dataset seimbang dan tidak seimbang berdasarkan percobaan yang dilakukan menghasilkan rata-rata performa klasifikasi seperti pada Tabel 4.10, dimana setiap skenario dijalankan sebanyak 3 (tiga) kali untuk memperkuat hasil analisa. Proses monitoring pada tahapan *training* dilakukan dengan teknik monitoring *EarlyStopping*.

Pengukuran kinerja pada model dengan kondisi dataset yang telah mengalami *oversampling* dinilai berdasarkan perolehan nilai rata-rata akurasi, kinerja terbaik pada model FastText dicapai saat menerapkan parameter *window*=5 dan *embedding size*=300 dengan nilai akurasi sebesar 87,85%. Sama seperti kondisi data seimbang, pada kondisi tidak seimbang seperti yang disajikan pada Tabel 4.7 menunjukkan kinerja terbaik diperoleh pada model FastText dengan parameter nilai *window*=5 dan *embedding size*=300, meskipun pada kondisi ini pengukuran kinerja dilakukan berdasarkan nilai *f1-score* yakni dengan perolehan sebesar 86,29%.

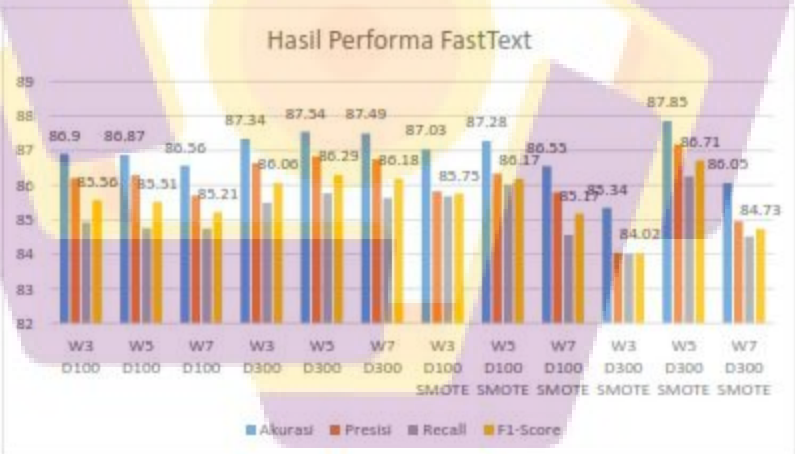
Tabel 4. 10. Hasil Performa FastText

Word Embedding	Data	Window			Dimensi		Rata-rata Performa			
		3	5	7	100	300	Akurasi	Presisi	Recall	F1-Score
FastText	Tidak Seimbang (Original)	√	-	-	√	-	86,90	86,20	84,92	85,56
		-	√	-	√	-	86,87	86,28	84,75	85,51
		-	-	√	√	-	86,56	85,69	84,74	85,21
		√	-	-	-	√	87,34	86,64	85,49	86,06
		-	√	-	-	√	87,54	86,83	85,76	86,29
		-	-	√	-	√	87,49	86,75	85,62	86,18

Tabel 4. 10. Hasil Performa FastText (Lanjutan)

Word Embedding	Data	Window			Dimensi		Rata-rata Performa			
		3	5	7	100	300	Akurasi	Presisi	Recall	F1-Score
	Scimbang (SMOTE)	√	-	-	√	-	87,03	85,82	85,67	85,75
		-	√	-	√	-	87,28	86,33	86,01	86,17
		-	-	√	√	-	86,55	85,79	84,56	85,17
		√	-	-	-	√	85,34	84,03	84,01	84,02
		-	√	-	-	√	87,85	87,17	86,25	86,71
		-	-	√	-	√	86,05	84,95	84,51	84,73

Perbandingan perolehan hasil performa penerapan *word embedding* FastText memiliki selisih perbedaan yang tidak terlalu siginifikan. Hal ini dapat dilihat pada grafik hasil pengujian dengan variasi parameter FastText yang divisualisasikan pada Gambar 4.9.



Gambar 4. 9. Perbandingan Kinerja Variasi Parameter FastText - Bi-LSTM

FastText mampu menangani kata-kata dengan morfologi yang kompleks karena mempelajari representasi untuk sub-kata dengan pemodelan *subword* (*n*-

gram). Sehingga representasi kata-kata yang dihasilkan memiliki hasil yang baik untuk kata-kata yang memiliki frekuensi kemunculan sedikit atau bahkan tidak terdapat pada corpus pelatihan. Sementara karena selain adanya representasi kata-kata secara penuh yaitu representasi untuk sub-word mengakibatkan model FastText memiliki ukuran yang cenderung lebih besar dan berpotensi mengalami *overhead* komputasi.

4.7. Model GloVe

GloVe menggunakan metode faktorisasi matriks global yang merupakan matriks yang menunjukkan kemunculan kata-kata tertentu dalam suatu dokumen. GloVe juga mempelajari hubungan kata dengan menghitung seberapa sering kata muncul satu sama lain sehingga mampu mengidentifikasi hubungan semantik antar kata dalam korpus tertentu. Rasio probabilitas kemunculan kata berpotensi untuk menyandikan suatu bentuk makna dan membantu meningkatkan kinerja pada masalah analogi kata.

Setiap kata dihitung berdasarkan frekuensi kemunculan bersama, proses ini dilakukan menggunakan matriks co-occurence. Berikut ilustrasi *co-occurrence* secara sederhana yang diperoleh dari 3 (tiga) kalimat ulasan disajikan pada Tabel 4.11.

Selanjutnya dilakukan inisialisasi vector kata W dan W' serta bias b dan b' untuk inisialisasi bobot. Perhitungan nilai prediksi dilakukan dengan *forward pass*, hal ini juga dimanfaatkan untuk meminimalkan perbedaan antara nilai prediksi dan nilai sebenarnya dari *co-occurrence matrix* yaitu dengan fungsi *Loss*. Kemudian

nilai bobot dan bias akan diperbarui menggunakan *gradient descent* pada tahap *Backward Pass*.

Tabel 4. 11. Matriks Co-occurence GloVe

Kata	Saya	Udah	Mencoba	Berulang	Kali	Tetep	Gk	Muncul	Kode	Transaksinya	...
Saya	0	1	0	0	0	0	0	0	0	0	...
Udah	1	0	1	0	0	0	0	0	0	0	...
Mencoba	0	1	0	1	0	0	0	0	0	0	...
Berulang	0	0	1	0	1	0	0	0	0	0	...
Kali	0	0	0	1	0	1	0	0	0	0	...
Tetep	0	0	0	0	1	0	1	0	0	0	...
Gk	0	0	0	0	0	1	0	1	0	0	...
Muncul	0	0	0	0	0	0	1	0	1	0	...
Kode	0	0	0	0	0	0	0	1	0	1	...
Transaksinya	0	0	0	0	0	0	0	0	1	0	...
...	...	...	...	...	...	...	...	...	...	...	...

4.7.1. Pembuatan Model GloVe

Model *pre-trained* GloVe yang dibangun dilakukan dengan menggunakan varian 100 dimensi dan 300 dimensi. Proses pelatihan dilakukan dengan memanfaatkan file glove.6B.100d dan glove.6B.300d berektensi .txt yang berisi representasi vektor berbagai kata global yang telah dilatih oleh model GloVe sebelumnya, Model dilatih menggunakan kumpulan data dengan ukuran data besar berjumlah sekitar 6 miliar kata, model berisi kumpulan hasil kata yang dikonversikan menjadi vektor 100 dimensi atau 300 dimensi. Selanjutnya proses pelatihan model GloVe menghasilkan representasi vektor untuk kata-kata berdasarkan kemunculan dan hubungannya dengan kata-kata lain dalam konteks yang ditemukan dalam data latihan. Model GloVe diterapkan dengan memanfaatkan Google Colaboratory dengan mengimport file model *pre-trained*.

4.7.2. Training dan Testing Glove

Proses *training* dan *testing* model GloVe tidak jauh berbeda dengan proses model *word embedding* sebelumnya. Hanya saja proses pelatihan pada model ini dilakukan menggunakan model yang telah dilatih sebelumnya. Yakni dengan meng-*import* file berisi hasil vektorisasi model GloVe berdimensi 100 dan 300 pada Google Colaboratoty. Teknik monitoring yang digunakan dalam pelatihan model dilakukan dengan metode *EarlyStopping*. Proses preprocessing juga diterapkan dalam model GloVe. Seperti Word2Vec dan FastText, setiap skenario GloVe juga dijalankan sebanyak 3 (tiga) kali untuk memperkuat hasil analisa. Kombinasi penerapan parameter model GloVe dan kondisi dataset seimbang dan tidak seimbang berdasarkan percobaan yang dilakukan menghasilkan performa rata-rata klasifikasi seperti pada Tabel 4.12.

Pengukuran kinerja pada model dengan kondisi dataset yang telah mengalami *oversampling* dinilai berdasarkan perolehan nilai akurasi, kinerja terbaik pada model GloVe dicapai saat menerapkan parameter *embedding size*=300 dengan nilai akurasi sebesar 85,98%. Sama seperti kondisi data seimbang, pada kondisi tidak seimbang seperti yang disajikan pada Tabel 4.8 menunjukkan kinerja terbaik diperoleh pada model GloVe dengan parameter *embedding size*=300, meskipun pada kondisi ini pengukuran kinerja dilakukan berdasarkan perolehan nilai *f1-score* yakni dengan perolehan sebesar 84,54%. Pada kedua kondisi menunjukkan penerapan *embedding size* sebesar 300 mampu lebih unggul dibandingkan penerapan *embedding size* 100.

Tabel 4. 12. Hasil Performa GloVe

Word Embedding	Data	Dimensi		Rata-rata Performa			
		100	300	Akurasi	Presisi	Recall	F1-Score
GloVe	Tidak Scimbang (Original)	√	-	85,52	85,28	82,63	83,93
		-	√	86,02	86,18	82,97	84,54
	Scimbang (SMOTE)	√	-	85,52	84,70	83,49	84,09
		-	√	85,98	84,96	84,15	84,55

Perbandingan perolehan hasil performa penerapan *word embedding* GloVe memiliki selisih perbedaan yang tidak terlalu siginifikan. Hal ini dapat dilihat pada grafik hasil pengujian dengan variasi parameter GloVe yang divisualisasikan pada Gambar 4.10.

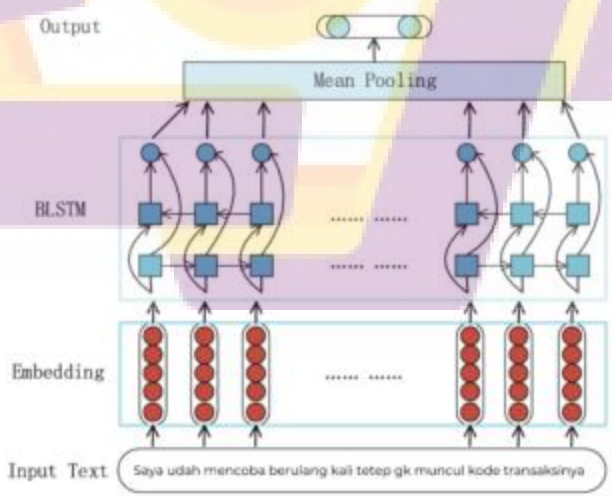
Gambar 4. 10. Perbandingan Kinerja Variasi Parameter GloVe - Bi-LSTM

Representasi yang dihasilkan oleh GloVe dinilai mencerminkan hubungan semantic yang baik karena dilakukan dengan memanfaatkan informasi *co-occurrence* korpus secara global. Hal ini menghasilkan representasi yang dekat dan berkualitas untuk kata-kata yang memiliki frekuensi kemunculan bersama yang tinggi. Meskipun seperti itu, GloVe tidak mampu secara efektif menangani kata-kata baru maupun kata-kata yang langka karena tidak dapat memodelkan sub-kata,

serta GloVe membutuhkan memori lebih untuk membangun dan menyimpan matriks *co-occurrence* yang besar.

4.8. Model Klasifikasi

Tahapan klasifikasi yang dilakukan dalam penelitian ini dilakukan dengan menerapkan model Bi-LSTM. Model klasifikasi melatih dan menguji data ulasan dengan menggunakan vector-vektor yang dihasilkan dari proses fitur ekstraksi dengan *word embedding* seperti Word2Vec, FastText, dan GloVe. Seperti yang dapat dilihat pada Gambar 4.11, setiap data ulasan berperan sebagai input yang akan direpresentasikan menjadi vector berdimensi sebagai output dari *Embedding Layer*. Selanjutnya data akan diproses pada beberapa layer Bidirectional LSTM, hingga pada tahapan akhir menuju *Dense Layer* yang berperan sebagai *output layer*. Pada *output layer* inilah akan diperoleh hasil klasifikasi analisis sentiment, yaitu positif atau negatif.



Gambar 4. 11. Model Klasifikasi Analisis Sentimen

Output yang dihasilkan pada tahapan *embedding* digunakan sebagai input pada hidden layer pertama. Proses diawali dengan fungsi LSTM pada Persamaan 10.

$$y' = a\left(\sum_{i=1}^{n} W_{ij} X_j + b\right) \quad (10)$$

Dimana *y'* atau *oh* merupakan output dari LSTM, *a* merupakan fungsi aktivasi, *n* merupakan banyaknya data, *W* merupakan bobot, dan *b* merupakan bias.

Secara sederhana, output neuron dihitung dengan perkalian input dengan bobot dan nilai bias. Kemudian, fungsi aktivasi digunakan untuk menghitung hasilnya. Fungsi aktivasi *tanh*, yang merupakan fungsi aktivasi LSTM standar yang digunakan dalam penelitian ini, memiliki luaran yang berkisar dari -1 hingga 1. Fungsi aktivasi tanh dijabarkan pada Persamaan 11.

$$F(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (11)$$

Output dari neuron pada *hidden layer* pertama akan menjadi masukan untuk neuron pada *hidden layer* kedua, yang kemudian dikalikan dengan masing-masing bobot. Begitu seterusnya hingga output pada *hidden layer* terkahir digunakan sebagai input untuk dense layer, yang merupakan hasil prediksi model terhadap teks ulasan.

Kemudian diperoleh nilai error seperti pada Persamaan 12 berdasarkan perbandingan antara output yang seharusnya dengan hasil prediksi. Nilai *error* inilah yang digunakan untuk perbaikan nilai bobot dan bias pada setiap layer yang sebelumnya ditentukan secara acak oleh model. Dengan adanya *Bidirectional layer* proses pembelajaran (*update* nilai bobot dan bias) yang didasarkan dari nilai *error* dilakukan secara dua arah, sehingga update nilai menghasilkan *output* yang

semakin mendekati hasil yang seharusnya. Adapun proses *update* bobot dan bias dilakukan dengan Persamaan 13 dan Persamaan 14.

$$E = y - y' \tag{12}$$

$$W' = W + (lr \times E \times X) \tag{13}$$

$$b' = b + (lr \times E) \tag{14}$$

Dimana W merupakan bobot dan W' bobot baru yang telah di*update*, b merupakan bias dan b' nilai bias baru, lr merupakan *learning rate*, E merupakan nilai selisih antara output dan hasil yang seharusnya, dan X merupakan input sebelum diperbarui.

Berikut ilustrasi perhitungan klasifikasi model Bi-LSTM pada *layer output* yang dilakukan terhadap data ulasan pada Tabel 4.13 yang terdiri atas 2 (dua) kelas, yaitu positif dan negatif.

Tabel 4. 13. Contoh Kalimat Ulasan

Ulasan 1	Mode hemat jg sangat membantu saat kantong tipis
Ulasan 2	Saya udah mencoba berulang kali tetep gk muncul kode transaksinya
Ulasan 3	Performa makin lama makin menurun dan harga yg di tawarkan juga lebih tinggi

Sebelum digunakan pada model klasifikasi, data ulasan melalui tahapan *preprocessing* hingga diperoleh hasil seperti pada Tabel 4.14.

Tabel 4. 14. Contoh Kalimat Ulasan Setelah Preprocessing

No	Ulasan										Label
	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	
1	mode	hemat	juga	sangat	bantu	saat	kantong	tipis			Positif
2	saya	sudah	coba	ulang	kali	tetap	tidak	muncul	kode		Negatif
3	performa	makin	lama	makin	turun	harga	tawar	lebih	tinggi		Negatif

1. Menentukan bobot dan bias

   Neuron 1 *output layer*:

   Bobot: $w_{1,1} = 0.1$, $w_{1,10} = 0.15$

   Bias: $b_{k1} = 0.6$

   Neuron 2 *output layer*:

   Bobot: $w_{2,1} = 0.1$, $w_{2,10} = 0.15$

   Bias: $b_{k2} = 0.67$

2. Menghitung aktivasi neuron untuk setiap ulasan

   Ulasan 1: "mode hemat juga sangat bantu saat kantong tipis" dengan label "Positif".

   Input vector (x_1 hingga x_{10}): $w_{1,1} = 0.1$, $w_{1,10} = 0.15$ (sesuai dengan nilai yang telah ditentukan).

   Menghitung untuk Neuron 1:

   $$oh_1 = \tanh (w_{1,1} \times x_1 + \ldots + w_{1,10} \times x_{10} + b_{k1})$$

   $$oh_1 = \tanh (0.1 \times 0.1 + \ldots + 0.15 \times 0.1 + 0.6)$$

   $$oh_1 = \tanh (0.1 + \ldots + 0.0225 + 0.6)$$

   $$oh_1 = \tanh (0.6325)$$

   $$oh_1 = 0.5605$$

   Menghitung untuk Neuron 2:

   $$Oh_2 = \tanh (w_{2,1} \times x_1 + \ldots + w_{2,10} \times x_{10} + b_{k2})$$

   $$oh_1 = \tanh (0.1 \times 0.1 + \ldots + 0.15 \times 0.1 + 0.67)$$

   $$oh_1 = \tanh (0.1 + \ldots + 0.0225 + 0.67)$$

   $$oh_1 = \tanh (0.7025)$$

$oh_1 = 0.6053$

3. Menghitung *Softmax*

$$Y'_{softmax,i} = \frac{e^{oh_i}}{\sum_j e^{oh_j}} \quad (15)$$

Dimana *i* merupakan indeks dari neuron output layer (1, 2, atau 3).

Berikut contoh perhitungan *softmax*.

$e^{oh1} = e^{0.5605} \rightarrow 1.751$

$e^{oh2} = e^{0.6053} \rightarrow 1.832$

$e^{oh1} + e^{oh2} + e^{oh3} = 3.583$

$$Y'_{softmax,1} = \frac{1.751}{3.583} = 0.489$$

$$Y'_{softmax,2} = \frac{1.832}{3.583} = 0.511$$

4. Menentukan klasifikasi akhir

Dalam menentukan klasifikasi akhir dilakukan berdasarkan nilai *argmax* dari nilai *softmax*. Adapun nilai *argmax* merupakan nilai maksimal *softmax* dari seluruh neuron sebagaimana pada Persamaan 16. Nilai *softmax* terbesar menunjukan klasifikasi kelas dari ulasan.

$$Y'argmax = \max(Y'_{softmax,i}) \quad (16)$$

Sehingga pada ulasan pertama memiliki nilai *argmax* = 0.511 yaitu yang terletak pada Neuron *output layer* 2. Nilai ini menunjukan bahwa ulasan pertama diklasifikasikan dalam kelas Negatif.

5. Membandingkan dengan label sebenarnya

Setelah ulasan berhasil diklasifikan selanjutnya dilakukan pengecekan dengan membandingkan kelas hasil klasifikasi dengan kelas

sebenarnya. Hasil perbandingan disimbolkan dengan T=*True* jika kelas hasil klasifikasi sama dengan kelas sebenarnya atau F=*False* jika kelas antara hasil klasifikasi dan kelas sebenarnya berbeda.

Pada ulasan pertama hasil klasifikasi yang dihasilkan adalah kelas Negatif, sementara ulasan memiliki kelas sebenarnya adalah Positif, sehingga untuk ulasan pertama disimbolkan dengan F (*False*).

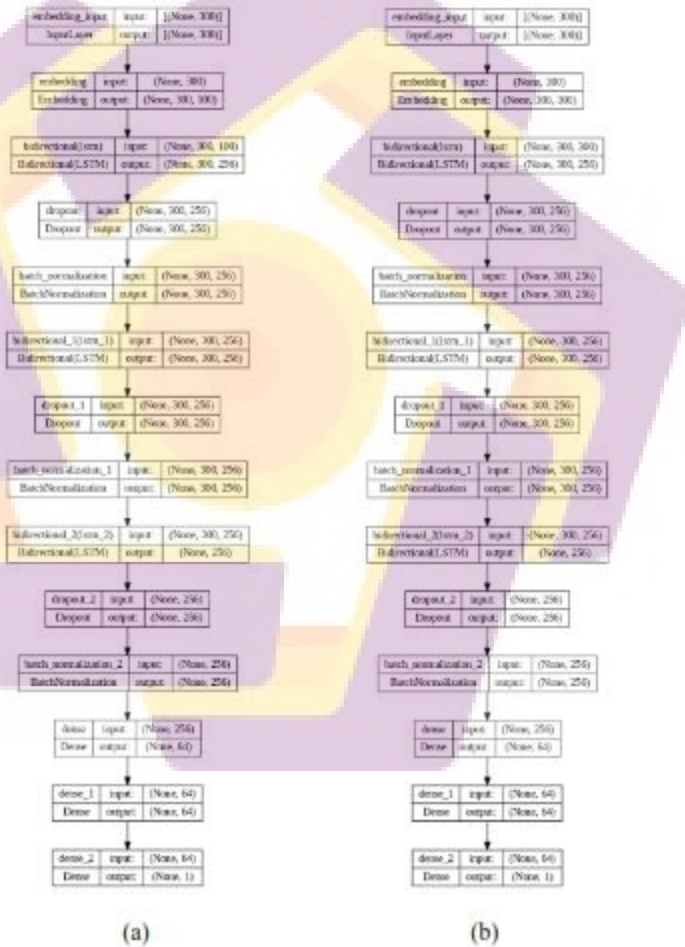
Berdasarkan ulasan yang digunakan sebagai contoh ilustrasi perhitungan klasifikasi dengan Bi-LSTM, 2 (dua) ulasan mampu diklasifikasikan pada kelas sebenarnya dan 1 (satu) ulasan diklasifikasikan pada kelas yang bukan sebenarnya. Adapun hasil perhitungan setiap tahapan Bi-LSTM secara singkat disajikan pada Tabel 4.15.

Tabel 4. 15. Contoh Hasil Perhitungan Klasifikasi Bi-LSTM

No	Ulasan	Output Layer		Y' Softmax	Y' Argmax	Y	R
		Neuron 1	Neuron 2				
1	Mode hemat jg sangat membantu saat kantong tipis	0.5605 tanh (0.6325)	0.6053 (tanh: 0.7025)	[0.489, 0.511]	[0, 1]	[1, 0]	False
2	Saya udah mencoba berulang kali tetep gk muncul kode transaksinya	0.46 tanh (0.5)	0.54 tanh (0.6)	[0.480, 0.519]	[0, 1]	[0, 1]	True
3	Performa makin lama makin menurun dan harga yg di tawarkan juga lebih tinggi	0.38 tanh (0.4)	0.72 tanh (0.9)	[0.415 ,0.584]	[0, 1]	[0, 1]	True

Model klasifikasi yang digunakan merupakan sebuah model *Sequential* dalam Keras yang terdiri dari beberapa layer yang berbeda, dengan arsitektur yang melibatkan penggunaan LSTM (*Long Short-Term Memory*) yang berbasis pada

arsitektur BiLSTM (*Bidirectional Long Short-Term Memory*) dengan *hidden unit* 256 untuk mengolah data teks seperti pada Gambar 4.12, (a) model embedding berdimensi 100; (b) model embedding berdimensi 300. Vektor-vektor yang telah dibangun pada tahapan fitur ekstraksi dengan *word embedding* berdasarkan dataset digunakan sebagai *training* dan *testing* model klasifikasi ini.



(a) (b)

Gambar 4. 12. Model BiLSTM-Embedding (a)100 dimensi; (b)300 dimensi

Layer Embedding, merupakan layer pertama dalam model. Bertanggung jawab untuk mengonversi input kata-kata ke dalam representasi vektor (*embedding vectors*). Input Shape: (None, 300), artinya menerima input sepanjang 300 kata. Output Shape: (None, 300, 100) atau (None, 300, 300), menghasilkan representasi vektor 100 atau 300 dimensi untuk setiap kata dalam input. Layer ini bertanggung jawab untuk mengonversi input teks menjadi representasi vektor dengan dimensi tertentu (dalam kasus ini, 100 dan 300 dimensi). *Embedding* digunakan untuk merepresentasikan kata-kata dalam teks ke dalam bentuk vektor yang dapat dimengerti oleh model. Hal ini memungkinkan model untuk mempelajari hubungan antara kata-kata berdasarkan konteksnya.

Layer Bidirectional LSTM, menggunakan dua LSTM yang beroperasi secara simultan: satu memproses urutan masukan dalam urutan maju dan yang lainnya memprosesnya dalam urutan mundur. *Input Shape*: (None, 300, 300), menerima *input* dari *layer embedding*. *Output Shape*: (None, 300, 256), menghasilkan urutan output LSTM dengan panjang 300 dan dimensi 256. Penggunaan BiLSTM memungkinkan model untuk memahami konteks kata tidak hanya dari kata-kata sebelumnya dalam urutan teks, tetapi juga dari kata-kata setelahnya. Hal ini membantu model untuk menangkap hubungan temporal yang kompleks dalam teks.

Layer Dropout, merupakan layer yang digunakan untuk mencegah *overfitting* dengan mengatur tingkat dropout pada *output layer* sebelumnya, dilakukan secara acak mengabaikan sebagian unit (*neuron*) selama proses pelatihan. *Input Shape*: (None, 300, 256), menerima input dari *layer* LSTM. *Output*

Shape: Sama dengan *input shape*. Dropout membantu mencegah model dari mengandalkan terlalu banyak pada fitur spesifik dalam data pelatihan, sehingga meningkatkan generalisasi model pada data baru.

Layer Batch Normalization, layer ini bertanggung jawab untuk mempercepat konvergensi selama pelatihan dan mengurangi *overfitting* dengan menormalisasi output dari layer sebelumnya. *Input Shape*: (None, 300, 256), menerima input dari layer Dropout. *Output Shape*: Sama dengan input shape. Normalisasi batch membantu mengurangi perubahan distribusi input ke layer selanjutnya selama pelatihan, yang dapat membantu dalam pelatihan yang lebih stabil dan konvergensi yang lebih cepat.

Layer Dense (*fully connected*), layer ini adalah layer tersembunyi yang terdiri dari neuron-neuron yang sepenuhnya terhubung. *Input Shape*: (None, 300, 40), menerima input dari layer *Batch Normalization*. *Output Shape*: (None, 64), artinya menghasilkan output dengan dimensi 64. Selain itu terdapat Layer Dense (*fully connected*) yang merupakan layer Dense kedua. *Input Shape*: (None, 64), menerima input dari layer Dense sebelumnya. *Output Shape*: (None, 64), artinya menghasilkan output dengan dimensi 64. Layer Dense digunakan untuk melakukan pembelajaran pola yang lebih kompleks dari representasi vektor yang diberikan oleh layer sebelumnya.

Layer Dense (*fully connected*) ketiga yang juga berperan sebagai Layer Dense *Output*. *Input Shape*: (None, 64), menerima input dari layer Dense sebelumnya. *Output Shape*: (None, 1), artinya menghasilkan output dengan dimensi tunggal (digunakan untuk klasifikasi biner, misalnya). Layer ini menghasilkan

output dari model, dalam kasus ini adalah 1 dimensi karena model ini digunakan untuk tugas klasifikasi biner (yaitu untuk menentukan sentimen analisis). Layer Dense output digunakan untuk melakukan klasifikasi pada representasi vektor yang telah dipelajari oleh model, menghasilkan prediksi berdasarkan informasi yang diambil dari seluruh teks yang dimasukkan ke dalam model.

4.9. Pembahasan Hasil

4.9.1. Analisis Hasil Klasifikasi

Total seluruh percobaan yang dilakukan dalam penelitian ini berjumlah 84 kali, masing-masing percobaan terdiri atas penerapan model *word embedding* dengan berbagai skenario. Skenario penerapan *word embedding* Word2Vec berjumlah sebanyak 12 kali, FastText sebanyak 12 kali, dan GloVe sebanyak 4 kali. Dimana setiap skenario dijalankan sebanyak 3 (tiga) kali untuk memperkuat hasil analisa. Kemudian hasil tahapan fitur ekstraksi dengan *word embedding* tersebut diklasifikasikan dengan menerapkan model Bi-LSTM. Hasil pelatihan dan pengujian dari keseluruhan skenario secara lengkap dijabarkan pada Tabel 4.16.

Pengukuran kinerja yang dihasilkan model bergantung pada ada atau tidaknya ketidakseimbangan yang signifikan pada kumpulan data yang digunakan. Pada kondisi data yang tidak seimbang akurasi tidak lagi menjadi ukuran yang dapat diandalkan karena memberikan perkiraan yang terlalu optimis mengenai pengklasifikasi kemampuan di kelas mayoritas, sehingga kinerja yang dinilai pada kondisi ini adalah *f1-score* (Gu, Zhu, & Cai, 2009).

Pada kondisi dataset yang tidak seimbang performa terbaik dihasilkan saat penerapan *word embedding* FastText yaitu dengan perolehan rata-rata *f1-score* sebesar 86,29%, hal ini dihasilkan pada percobaan dengan nilai *window*=5, dan *embedding size*= 300. Pada kondisi dataset yang telah mengalami *oversampling* dengan SMOTE performa terbaik dihasilkan saat penerapan *word embedding* FastText dengan perolehan nilai akurasi sebesar 87,85%, perolehan ini dihasilkan pada percobaan dengan nilai *window*=5 dan *embedding size*=300, sedikit lebih unggul dibandingkan Word2Vec pada kondisi nilai *wondow*=5 dan *embedding size*=300 dengan selisih akurasi 0,34%.

Tabel 4. 16. Hasil Performa Model Klasifikasi Analasis Sentimen

Word Embedding	Data	Window			Dimensi		Akurasi	Presisi	Recall	F1-Score
		3	5	7	100	300				
Word2Vec	Tidak Seimbang (Original)	√	-	-	√	-	87,01	86,50	84,83	85,66
		-	√	-	√	-	87,25	86,75	85,13	85,93
		-	-	√	√	-	86,97	86,39	84,86	85,61
		√	-	-	-	√	86,52	85,75	84,55	85,15
		-	√	-	-	√	87,34	86,87	85,20	86,03
		-	-	√	-	√	87,07	86,31	85,23	85,77
	Seimbang (SMOTE)	√	-	-	√	-	87,02	86,05	85,43	85,74
		-	√	-	√	-	87,26	86,34	85,73	86,04
		-	-	√	√	-	86,97	85,86	85,22	85,54
		√	-	-	-	√	86,12	84,87	84,91	84,89
		-	√	-	-	√	87,51	86,58	85,99	86,28
		-	-	√	-	√	86,90	86,05	85,27	85,66
FastText	Tidak Seimbang (Original)	√	-	-	√	-	86,90	86,20	84,92	85,56
		-	√	-	√	-	86,87	86,28	84,75	85,51
		-	-	√	√	-	86,56	85,69	84,74	85,21
		√	-	-	-	√	87,34	86,64	85,49	86,06
		-	√	-	-	√	87,54	86,83	85,76	86,29
		-	-	√	-	√	87,49	86,75	85,62	86,18

Tabel 4. 16. Hasil Performa Model Klasifikasi Analasis Sentimen (Lanjutan)

Word Embedding	Data	Window			Dimensi		Akurasi	Presisi	Recall	F1-Score
		3	5	7	100	300				
	Seimbang (SMOTE)	√	-	-	√	-	87,03	85,82	85,67	85,75
		-	√	-	√	-	87,28	86,33	86,01	86,17
		-	-	√	√	-	86,55	85,79	84,56	85,17
		√	-	-	-	√	85,34	84,03	84,01	84,02
		-	√	-	-	√	87,85	87,17	86,25	86,71
		-	-	√	-	√	86,05	84,95	84,51	84,73
GloVe	Tidak Seimbang (Original)	-	-	-	√	-	85,52	85,28	82,63	83,93
		-	-	-	-	√	86,02	86,18	82,97	84,54
	Seimbang (SMOTE)	-	-	-	√	-	85,52	84,70	83,49	84,09
		-	-	-	-	√	85,98	84,96	84,15	84,55

Setiap penerapan masing-masing model *word embedding* memiliki kondisi unggul pada parameter yang berbeda. Sementara berdasarkan rata-rata performa yang diperoleh dari pengujian yang dilakukan sebanyak 3 (tiga) kali untuk setiap skenarionya, performa model terbaik diperoleh pada penerapan *window*=5 dan *embedding size*=300 baik pada kondisi data tidak seimbang maupun pada kondisi data yang telah mengalami *oversampling*, hal ini berlaku pada penerapan ketiga jenis *word embedding* (Word2Vec, FastText, dan GloVe).

Secara keseluruhan, model dengan *word embedding* FastText mampu unggul pada semua kondisi data, yakni dengan perolehan f1-score sebesar 86,29% pada kondisi data tidak seimbang dan akurasi sebesar 87,85% pada kondisi data setelah *oversampling*. Kemudian model dengan *word embedding* Word2Vec dan GloVe memperoleh f1-score sebesar 86,03% dan 84,54% pada kondisi data tidak seimbang serta akurasi sebesar 87,51% dan 85,98% pada kondisi data setelah

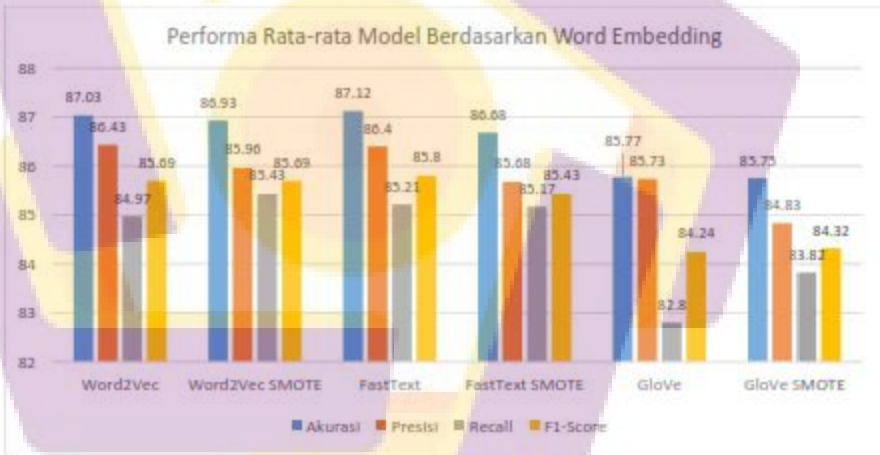
oversampling. Perbandingan hasil performa terbaik tiap model penerapan *word embedding* disajikan pada Gambar 4.13.

Gambar 4. 13. Perbandingan Performa Terbaik Word Embedding

Selain analisa berdasarkan skenario dengan parameter berbeda, performa model juga diukur dan dibandingkan berdasarkan masing-masing jenis *word embedding* dan kondisi data pelatihan (Gambar 4.14 dan Tabel 4.17). Secara rata-rata berdasarkan jenis *word embedding* dan kondisi data, model FastText mampu memperoleh performa yang paling unggul pada kondisi data original (*imbalanced dataset*), diikuti oleh performa model Word2Vec kemudian GloVe. Hal yang sama pada kondisi dataset yang telah mengalami *oversampling*, model FastText juga mampu lebih unggul dibandingkan model Word2Vec dan GloVe.

Tabel 4. 17. Rata-rata Performa Model Berdasarkan *Word Embedding*

Word Embedding	Data	Performa Rata-rata			
		Akurasi	Presisi	Recall	F1-Score
Word2Vec	Tidak Seimbang (Original)	87,03	86,43	84,97	85,69
	Seimbang (SMOTE)	86,93	85,96	85,43	85,69
FastText	Tidak Seimbang (Original)	87,12	86,40	85,21	85,80
	Seimbang (SMOTE)	86,68	85,68	85,17	85,43
GloVe	Tidak Seimbang (Original)	85,77	85,73	82,80	84,24
	Seimbang (SMOTE)	85,75	84,83	83,82	84,32



Gambar 4. 14. Performa Rata-rata Model Berdasarkan *Word Embedding*

Sehingga dapat disimpulkan bahwa model Bi-LSTM dengan *word embedding* FastText mampu unggul dibandingkan Word2Vec dan GloVe baik terhadap penilaian berdasarkan parameter yang berbeda maupun secara rata-rata. Meskipun demikian dalam proses klasifikasi masih terdapat beberapa data yang

belum tepat diklasifikasikan pada kelas yang sebenarnya, hal ini disebabkan oleh beberapa faktor sebagai berikut.

1. Konteks yang ambigu, dalam suatu kalimat ulasan terdapat lebih dari satu makna atau perasaan. Seperti "Semakin Update versi semakin memperbanyak Peraturan.. Bagus sih..!! Tapi Untuk Aplikator .. Saya kasihan dengan Mitra kalian yang Rebahan di jalan arau bahkan Meluangkan waktu nya sampai Larut Malam hanya untuk Berkontribusi dengan aplikator.. Kalian pantau lah mereka..!! Jangan hanya memperbanyak biaya ini dan itu yang membuat semakin banyak Para Consumen enggan untuk menggunakan aplikasi ini, Yang berdampak Ke Mitra kalian secara Real time Yyang onbit hingga larut malam"

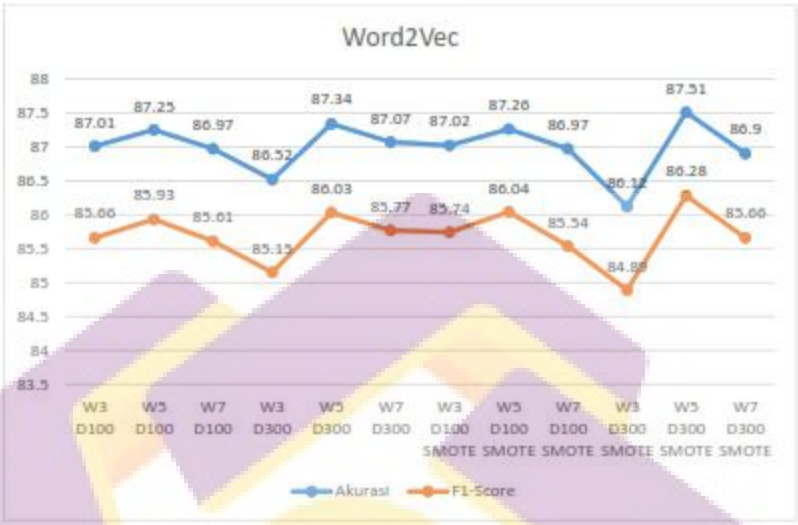
   Ulasan tersebut mengungkapkan makna positif terkait update aplikasi, namun kalimat selanjutnya berisi ungkapan yang bermakna negatif.
2. Penggunaan kalimat sarkasme dalam ulasan. Seperti "Aplikasi yg sangat bagus sekali, saking bagus nya mau mesen makanan aja harus buka tutup aplikasi berkali", jgn alasan pastikan sinyal kamu stabil, youtube an 1080 lancar buka aplikasi tokai ini muter", betul" Aplikasi yg patut di apresiasi kalo ada bintang minus 100 ku kasihin semua"

   Ulasan tersebut menggunakan kata "bagus sekali" yang bermakna sindiran atau kebalikannya, serta kata "patut diapresiasi" yang bertolakbelakang dengan "bintang minus 100", sehingga terjadi mispersepsi antara mesin dengan penilaian pengguna.

4.9.2. Analisis Performa Word Embedding

Setiap model klasifikasi dalam skenario penelitian ini menerapkan fitur ekstraksi berupa *word embedding* dengan model dan parameter yang beragam, seperti yang disajikan pada Tabel 4.2. Selain parameter yang disebutkan pada Tabel 4.2, diterapkan juga parameter lain seperti arsitektur CBOW, motode evaluasi *Hierarchical Softmax*, dan *minimum count* bernilai 10. Dengan perbedaan jenis atau nilai parameter yang diterapkan, setiap model menghasilkan performa yang beragam.

Model Word2Vec yang diterapkan pada kondisi data dengan kelas tidak seimbang menghasilkan performa *f1-score* antara 85,15% hingga 86,03%, sementara akurasi yang diperoleh pada kondisi data setelah *oversampling* menghasilkan performa antara 86,12% hingga 87,51%. Pada ilustrasi Gambar 4.15 performa terbaik Word3Vec ditunjukan pada parameter *window*=5, *embedding size*=300, dan kondisi data setelah *oversampling*. Sementara yang terendah ditunjukan pada parameter *window*=3, *embedding size*=300, dan kondisi data setelah *oversampling*. Pada kondisi data tidak seimbang maupun seimbang dengan *embedding size*=100, terjadi pola yang sama antara penerapan *window*=3, *window*=5, dan *window*=7. Dimana performa terbaik diperoleh pada penerapan *window*=5, selanjutnya *window*=3 kemudian *window*=7 dengan performa yang terendah. Sementara pada *embedding size*=300, pola yang dihasilkan adalah *window*=5 sebagai performa terbaik, diikuti oleh *window*=7, kemudian *window*=3 dengan performa terendah.



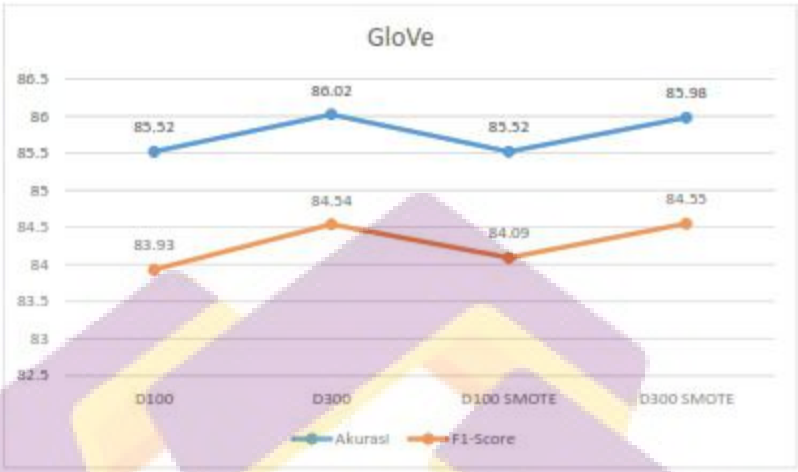
Gambar 4. 15. Analisis Performa Word2Vec

Model FastText pada kondisi dengan kelas tidak seimbang menghasilkan performa *f1-score* antara 85,21% hingga 86,29%, sementara akurasi yang diperoleh pada kondisi data setelah *oversampling* menghasilkan performa antara 85,34% hingga 87,85%. Pada ilustrasi Gambar 4.16 performa terbaik FastText ditunjukan pada parameter *window*=5, *embedding size*=300, dan kondisi data setelah *oversampling*. Sementara yang terendah ditunjukan pada parameter *window*=3, *embedding size*=300, dan kondisi data setelah *oversampling*. Pada kondisi data tidak seimbang maupun seimbang dengan *embedding size*=300, terjadi pola yang sama antara penerapan *window*=3, *window*=5, dan *window*=7. Dimana performa terbaik diperoleh pada penerapan *window*=5, selanjutnya *window*=7 kemudian *window*=3 dengan performa yang terendah. Sementara pada *embedding size*=100, terdapat perbedaan pola yang dihasilkan pada perbedaan kondisi data, sebelum

dilakukan *oversampling window*=3 menghasilkan performa terbaik, diikuti oleh *window*=5, kemudian *window*=7 dengan performa terendah, setelah oversampling *window*=5 justru mampu lebih baik dibandingkan *window*=3 dan *window*=7.

Gambar 4. 16. Analisis Performa FastText

Model GloVe pada kondisi data dengan kelas tidak seimbang menghasilkan performa *f1-score* terbaik sebesar 84,54% dan terendah sebesar 83,93%, sementara akurasi yang diperoleh pada kondisi data setelah *oversampling* menghasilkan performa terbaik sebesar 85,98% dan terendah sebesar 85,52%. Pada ilustrasi Gambar 4.17 performa terbaik GloVe ditunjukan pada parameter *embedding size*=300 dan kondisi data setelah *oversampling*. Sementara yang terendah ditunjukan pada parameter *embedding size*=100, dan kondisi data tidak seimbang. Pada kondisi data tidak seimbang maupun seimbang performa terbaik model GloVe dihasilkan pada penerapan *embedding size*=300.



Gambar 4. 17. Analisis Performa GloVe

Berdasarkan hasil pengujian, performa terbaik diperoleh pada parameter *window*=5 dan *embedding size*=300. Parameter *window* yang bertugas mengontrol dan memperhitungkan kata disekitar target saat pelatihan dapat mempengaruhi performa model. Dalam penyelesaian permasalahan pada penelitian ini *window* sebesar 5 merupakan yang paling mampu memberikan representasi kata yang kontekstual dan akurat. Saat nilai *window* yang digunakan terlalu besar seperti 7 atau atau terlalu kecil seperti 3, justru menghasilkan performa yang lebih rendah. Hal ini karena penggunaan window yang terlalu besar dapat mengakibatkan model menjadi terlalu rumit dan memerlukan lebih banyak sumber daya untuk pelatihan. Sebaliknya, *window* yang terlalu kecil mungkin tidak mampu mengumpulkan cukup informasi untuk memberikan representasi kata yang efektif.

Parameter lain yang diperhatikan dalam penelitian ini adalah dimensi atau *embedding size* model *word embedding*. Nilai dimensi (*embedding size*) yang

digunakan adalah 100 dan 300, baik pada model Word2Vec, FastText, maupun GloVe. Pada model Word2Vec. Secara umum, untuk ketiga model (Word2Vec, FastText, dan GloVe), meningkatkan ukuran embedding dapat membantu model menangani data bahasa yang lebih kompleks dengan lebih baik sehingga mampu meningkatkan kinerja model. Hal ini ditunjukkan dengan perolehan performa yang lebih unggul saat menerapkan nilai *embedding size* 300 dibandingkan 100. Meskipun demikian, peningkatan ini harus diimbangi dengan risiko *overfitting* dan kebutuhan akan sumber daya komputasi yang lebih besar. Sehingga diperlukan penentuan *embedding size* yang ideal dalam eksperimen, serta perlu disesuaikan dengan permasalahan dan data yang diselesaikan.

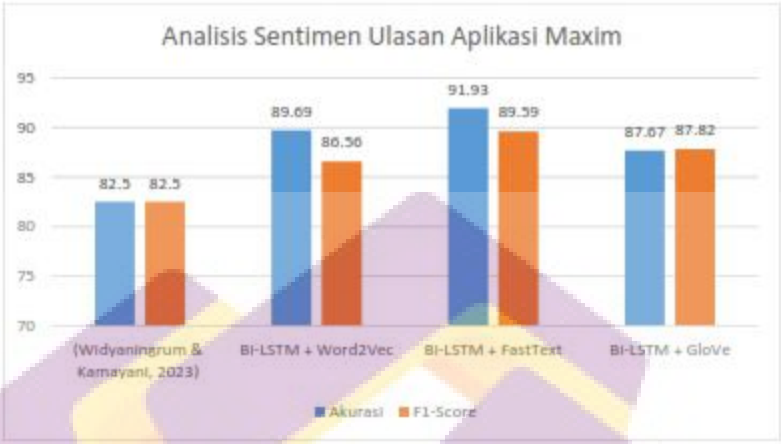
Word embedding FastText mampu unggul dibandingkan Word2Vec dan GloVe karena mampu melakukan pendekatan yang memanfaatkan sub-kata, yaitu dengan n-gram. Pendekatan ini memiliki keunggulan dalam menangani bahasa dengan morfologi yang kompleks, menangani kata-kata baru, hingga menangani kata-kata yang langka atau biasa dikenal *out of vocabulary* (OOV). Sementara Word2Vec dan GloVe cenderung memberikan representasi yang buruk untuk kata-kata yang jarang muncul atau tidak ada dalam korpus pelatihan karena mereka hanya mempelajari representasi kata-kata secara penuh. Selain itu, FastText mampu menghasilkan efisiensi yang baik dalam komputasi, sehingga cocok untuk diterapkan dalam berbagai aplikasi *Natural Language Processing* (NLP).

4.9.3. Perbandingan dengan Penelitian Terdahulu

Hasil performa yang diperoleh dalam penelitian ini memiliki hasil yang serupa dengan penelitian mengenai klasifikasi teks berita oleh (Nurdin, Aji, Bustamin, & Abidin, 2020) dimana penerapan *word embedding* Word2Vec dan FastText mampu menghasilkan performa yang unggul, terlebih saat dibandingkan dengan penerapan *word embedding* GloVe pada penelitian mengenai analisis sentimen ulasan amazon oleh (Alharbi, Alghamdi, Alkhammash, & Al Amri, 2021) dan penelitian mengenai deteksi emosi pada teks twitter oleh (Riza & Charibaldi, 2021).

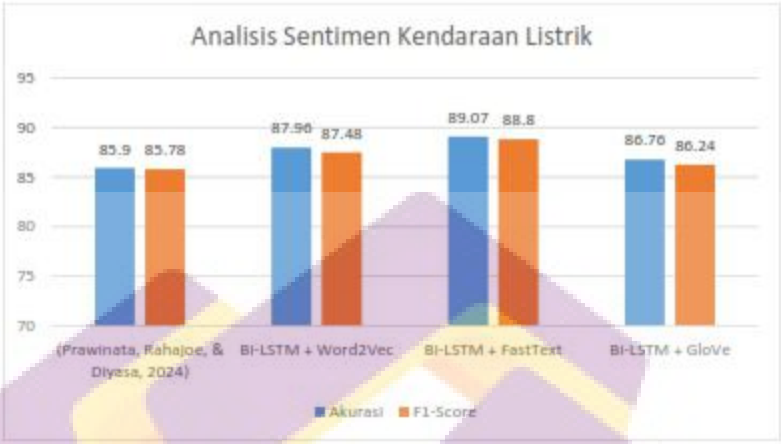
Model *word embedding* dengan performa terbaik dalam penelitian ini juga diterapkan untuk melakukan analisis sentimen menggunakan beberapa data penelitian terdahulu. Seperti data opini pengguna terhadap aplikasi Maxim (Widyaningrum & Kamayani, 2023) dan data opini masyarakat terkait kendaraan listrik (Prawinata, Rahajoe, & Diyasa, 2024). Data opini yang diterapkan merupakan data ulasan berbahasa Indonesia.

Ulasan terhadap aplikasi maxim terdiri atas 616 opini dengan sentimen negatif dan 526 opini dengan sentimen positif. Penerapan model menunjukan bahwa pengembangan model analisis sentimen yang dilakukan dengan penerapan Bi-LSTM dan *word embedding* mampu menghasilkan performa yang lebih ungul dibandingkan penerapan model analisis sentimen dengan algoritma *machine learning* yaitu naïve bayes pada penelitian (Widyaningrum & Kamayani, 2023). Hasil perbandingan performa model dalam penelitian ini dengan penelitian terdahulu disajikan pada Gambar 4.18.



Gambar 4. 18. Perbandingan Penelitian (Widyaningrum & Kamayani, 2023)

Perbandingan lain dilakukan dengan menerapkan model penelitian ini terhadap opini masyarakat mengenai kendaraan listrik sebanyak 30.000 data. Penelitian (Prawinata, Rahajoe, & Diyasa, 2024) menyarankan untuk melakukan eksplorasi terhadap penggunaan model LSTM yang lebih kompleks serta melakukan penanganan terkait data yang tidak seimbang dengan menerapkan SMOTE. Berdasarkan hasil uji coba, model Bi-LSTM dengan penerapan *word embedding* (Word2Vec, FastText, dan GloVe) dalam penelitian ini dinilai mampu menghasilkan performa yang lebih unggul dibandingkan LSTM meskipun dengan *embedding* yang sama yaitu Word2Vec. Perbandingan performa model disajikan pada Gambar 4.19.



Gambar 4. 19. Perbandingan Penelitian (Prawinata, Rahajoe, & Diyasa, 2024)

Kemudian diantara 3 (tiga) *word embedding* yang dibandingkan FastText mampu menghasilkan performa yang unggul dibandingkan Word2Vec dan GloVe baik pada data penelitian (Widyaningrum & Kamayani, 2023) maupun data penelitian (Prawinata, Rahajoe, & Diyasa, 2024), hasil tersebut serupa dengan eksperimen yang telah dilakukan dengan dataset Gojek dalam penelitian ini.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil eksperimen dalam penelitian analisis sentimen ulasan pengguna aplikasi Gojek pada situs Google Playstore dengan penerapan metode klasifikasi Bi-LSTM yang dikombinasikan dengan *word embedding*, diperoleh kesimpulan sebagai berikut.

1. Performa yang dihasilkan model klasifikasi memiliki hasil yang berbeda untuk setiap penerapan *word embedding*nya. Word2Vec dapat diterapkan pada corpus yang besar dengan menghasilkan representasi kata berkualitas tinggi karena menangkap hubungan semantic antar kata. FastText mampu menangani kata-kata yang memiliki morfologi kompleks maupun *out of vocabulary*, karena selain merepresentasikan kata secara penuh FastText juga memperhatikan sub-kata, yaitu dengan n-gram. GloVe memanfaatkan corpus global dengan matriks co-occurrence untuk menghasilkan representasi kata-kata secara umum.

   Berdasarkan hasil eksperimen, pada kondisi data original (*imbalanced data*), model Word2Vec, FastText, dan GloVe menghasilkan performa terbaik dengan perolehan masing-masing *f1-score* 86,03%, 86,29%, 84,54%, akurasi 87,34%, 87,54%, 86,02%, presisi 86,87%, 86,83%, 86,18%, dan recall 85,20%, 85,76%, 82,97%. Sementara pada kondisi data dengan kelas seimbang (setelah *oversampling*), model Word2Vec,

FastText, dan GloVe menghasilkan performa terbaik dengan perolehan masing-masing akurasi 87,51%, 87,85%, 85,98%, presisi 86,58%, 87,17%, 84,96%, recall 85,99%, 86,25%, 84,15%, dan *f1-score* 86,28%, 86,71%, 84,55%. Model FastText mampu unggul dibandingkan Word2Vec dan GloVe pada kondisi data tidak seimbang dan kondisi data setelah *oversampling* baik pada berbagai parameter maupun secara rata-rata karena pemodelan yang dilakukan dengan sub-kata mampu menghasilkan representasi yang lebih baik.

2. Penerapan *word embedding* Word2Vec, FastText, dan GloVe dengan berbagai parameter berbeda menghasilkan performa yang beragam. Pada penerapan Word2Vec dan FastText performa yang unggul diperoleh pada parameter *window*=5 dan *embedding size*=300, baik pada kondisi dataset original maupun dataset setelah *oversampling*. Begitu juga pada penerapan GloVe, performa terbaik diperoleh pada parameter *embedding size*=300 pada kondisi data original dan setelah *oversampling*.

   Parameter *window* berperan dalam mengontrol jumlah kata sekitar target saat proses pelatihan. Dalam penelitian ini, window sebesar 5 terbukti memberikan representasi kata yang paling kontekstual dan akurat. Nilai window yang terlalu besar atau kecil, seperti 7 atau 3, justru menghasilkan performa model yang lebih rendah karena dapat menyebabkan model menjadi terlalu rumit atau kurang informasi untuk representasi yang efektif. Selain itu dalam pemilihan *embedding size* harus disesuaikan dengan permasalahan dan data yang dihadapi, meskipun nilai embedding size yang

lebih besar mampu meningkatkan kinerja model Word2Vec, FastText, dan GloVe dalam menangani data bahasa yang kompleks. Namun, peningkatan ukuran embedding juga berisiko menyebabkan overfitting dan membutuhkan lebih banyak sumber daya komputasi. Adapun parameter lain yang diterapkan antara lain: arsitektur CBOW, metode evaluasi *hierarchical softmax*, dan *minimum count*=10.

5.2. Saran

Berdasarkan hasil eksperimen dan kesimpulan yang diperoleh dalam penelitian ini, maka dapat dikemukakan saran-saran sebagai berikut.

1. Penelitian lebih lanjut dapat dilakukan dengan eksplorasi lebih dalam berdasarkan parameter lain atau *hyperparameter tuning* dari *word embedding* Word2Vec, FastText, dan GloVe.
2. Model representasi teks *deep learning*, seperti BERT atau *Attention* dapat diterapkan sebagai pembanding dalam proses vektorisasi (konversi) teks.
3. Penerapan *stratified sampling* penelitian selanjutnya dapat dipertimbangkan untuk diterapkan setelah data mengalami kondisi yang seimbang.

DAFTAR PUSTAKA

PUSTAKA BUKU

Bhattacharjee, J. (2018). *fastText Quick Start Guide: Get started with Facebook's library for text representation and classification.* Birmingham: Packt Publishing Ltd.

Liu, B. (2020). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions.* United Kingdom: Cambridge University Press.

Poria, S., Hussain, A., & Cambria, E. (2018). *Multimodal Sentiment Analysis.* Socio-Affective Computing Vol. 8 Springer Nature.

Ravichandiran, S. (2021). *Getting Started with Google BERT: Build and Train State-of-the-art Natural Language Processing Models Using BERT.* United Kingdom: Packt Publishing.

Søgaard, A., Vulić, I., Ruder, S., & Faruqui, M. (2022). *Cross-Lingual Word Embeddings.* Switzerland: Springer International Publishing.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Abdelhady, N., Soliman, T. H., & Farghally, M. F. (2024). Stacked-CNN-BiLSTM-COVID: an effective stacked ensemble deep learning framework for sentiment analysis of Arabic COVID-19 tweets. *Journal of Cloud Computing: Advances, Systems and Applications, 13*(1), 85. doi:10.1186/s13677-024-00644-6

Af'idah, D. I., Handayani, S. F., & Pratiwi, W. R. (2021). Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen. *Jurnal Informatika: Jurnal pengembangan IT (JPIT), 6*(3), 156-161.

Al-Azani, S., & El-Alfy, E.-S. M. (2017). Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text. *The 8th International Conference on Ambient Systems, Networks and Technologies, ANT2017* (pp. 359-366). Procedia Computer Science 109.

Alghifari, D. R., Edi, M., & Firmansyah, L. (2022). Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia. *Jurnal Manajemen Informatika (JAMIKA)*, 89-99.

Alharbi, N. M., Alghamdi, N. S., Alkhammash, E. H., & Al Amri, J. F. (2021). Evaluation of Sentiment Analysis via Word Embedding and RNN Variants for Amazon Online Reviews. *Mathematical Problems in Engineering*, 1-10.

Aljedaani, W., Rustam, F., Ludi, S., Ouni, A., & Mkaouer, M. W. (2021). Learning Sentiment Analysis for Accessibility User Reviews. *2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)* (hal. 239-246). IEEE.

Alsharef, A., Aggarwal, K., Sonia, Koundal, D., Alyami, H., & Ameyed, D. (2022). An Automated Toxicity Classification on Social Media Using LSTM and Word Embedding. *Computational Intelligence and Neuroscience*.

Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharrya, U. R. (2021). ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. *Elsevier - Future Generation Computer Systems*.

Bhatia, S., Sharma, M., & Bhatia, K. K. (2017). Sentiment Analysis and Mining of Opinions. *Bhatia, S., Sharma, M., & Bhatia, K. K. (2017). Sentiment AnalyInternet of Things and Big Data Analytics Toward Next-Generation Intelligence*, 503-523.

Bhatia, S., Sharma, M., & Bhatia, K. K. (2018). Sentiment Analysis and Mining of Opinions. *Internet of Things and Big Data Analytics Toward Next - Generation Intelligence. Studies in Big Data, 30*, 503-523.

Bojanowski, P., Grave, E., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the association for computational linguistics, 5*, 135-146.

Cahyo, D. N., Farasalsabila, F., Lestari, V. B., Hanafi, Lestari, T., & Al Islami, F. R. (2023). Sentiment Analysis for IMDb Movie Review Using Support Vector Machine (SVM) Method. *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi, 8(2)*, 90-95.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kagelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16*, 321-357. doi:https://doi.org/10.1613/jair.953

Colón-Ruiz, C., & Segura-Bedmar, I. (2020). Comparing deep learning architectures for sentiment analysis on drug reviews. *Journal of Biomedical Informatics 110*.

Dale, R., Moisi, H., & Somers, H. (2000). *Handbook of Natural Language Processing (Editet by Robert Dale, Her*. CRC press.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1*. arXiv.

Dharma, E. M., Gaol, F. L., Warnars, H. L., & Soewito, B. (2022). The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (cnn) text classification. *Journal of Theoretical and Applied Information Technology, 100*(2), 349-359.

Gondhi, K. N., Chaahat, Sharma, E., Alharbi, A. H., Verma, R., & Shah, M. A. (2022). Efficient Long Short-Term Memory-Based Sentiment Analysis of E-Commerce Reviews. *Hindawi - Computational Intelligence and Neuroscience.*

Gu, Q., Zhu, L., & Cai, Z. (2009). Evaluation Measures of the Classification Performance of Imbalanced Data Sets. *Computational Intelligence and Intelligent Systems: 4th International Symposium, ISICA 2009* (pp. 461-417). Huangshi, China: Springer Berlin Heidelberg.

Haddi, E., Liu, X., & Shi, Y. (2013). The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science, 17*, 26-32.

Hidayatullah, A. F. (2016). Pengaruh Stopword Terhadap Peroforma Klasifikasi Tweet Berbahasa Indonesia. *JISKa, 1*(1), 1-4.

Iqbal, A., Amin, R., Iqbal, J., Alroobaea, R., Binmahfoudh, A., & Hussain, M. (2022). Sentiment Analysis of Consumer Reviews Using Deep Learning. *Sustainability*, 14(17), 10844.

Kaibi, I., Nfaoui, E. H., & Satori, H. (2019). A Comparative Evaluation of Word Embeddings Techniques for Twitter Sentiment Analysis. *2019 International conference on wireless technologies, embedded and intelligent systems (WITS)* (pp. 1-4). Fez, Morocco: IEEE. doi:10.1109/WITS.2019.8723864

Kharde, V. A., & Sonawane, S. S. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. *International Journal of Computer Applications, 139(11).*

Khoo, L. S., Bay, J. Q., Yap, M. L., Lim, M. K., Chong, C. Y., Yang, Z., & Lo, D. (2023). Exploring and Repairing Gender Fairness Violations in Word Embedding-based Sentiment Analysis Model through Adversarial Patches. *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)* (pp. 651-662). IEEE.

Koto, F., & Rahmaningtyas, G. Y. (2017). InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs. *International Conference on Asian Language Processing (IALP).*, (pp. 391-394). Singapore.

Liu, C., Zhang, P., Li, T., & Yan, Y. (2019). Semantic Features Based N-Best Rescoring Methods for Automatic Speech Recognition. *Applied Sciences, 9*(23), 5053. doi:doi:10.3390/app9235053

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. *Advances in neural information processing systems, 26.*

Mustikasari, D., Widaningrum, I., Arifin, R., & Putri, W. H. (2021). Comparison of Effectiveness of Stemming Algorithms in Indonesian Documents. *2nd Borobudur International Symposium on Science and Technology (BIS-STE 2020)* (pp. 154-158). Atlantis Press. doi:10.2991/aer.k.210810.025

Muttaqin, M. N., & Khasirudin, I. (2021). Analisis Sentimen Aplikasi Gojek Menggunakan Support Vector Machine dan K Nearest Neighbor. *UNNES Journal of Mathematics, 10*(2), 22-27.

Nugraha, F. A., Harani, N. H., & Habibi, R. (2020). *Analisis Sentimen Terhadap Pembatasan Sosial Menggunakan Deep Learning.* (R. M. Awangga, Ed.) Bandung: Kreatif Industri Nusantara.

Nurdeni, D. A., Budi, I., & Santoso, A. B. (2021). Sentiment Analysis on Covid19 Vaccines in Indonesia: From The Perspective of Sinovac and Pfizer. *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCI)*, (pp. 122-127). Surabaya, Indonesia.

Nurdin, A., Aji, B. A., Bustamin, A., & Abidin, Z. (2020). Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal TEKNOKOMPAK, Vol. 14, No. 2*, 74-79.

Onan, A. (2020). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency Computation Practice and Experience*.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 1532-15432). Doha, Qatar.

Prawinata, D. A., Rahajoe, A. D., & Diyasa, I. S. (2024). Analisis Sentimen Kendaraan Listrik Pada Twitter Menggunakan Metode Long Short Term Memory. *SABER: Jurnal Teknik Informatika, Sains dan Ilmu Komunikasi, 2*(1), 300-313. doi:10.59841/saber.v2i1.855

Putra, D. T., & Setiawan, E. B. (2023). Sentiment Analysis on Social Media with Glove Using Combination CNN and RoBERTa. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi, 7*(3), 457-563.

Putri, M. I., & Khasirudin, I. (2022). Analisis Sentimen Pengguna Aplikasi MarketplaceTokopediaPada Situs Google Play Menggunakan Metode Support Vector Machine(SVM), Naïve Bayes, dan Logistic Regression. *PRISMA, Prosiding Seminar Nasional Matematika, 5*, (pp. 759-766).

Rahman, A., Utami, E., & Sudarmawan. (2021). Sentimen Analisis Terhadap Aplikasi pada Google Playstore Menggunakan Algoritma Naive Bayes dan Algoritma Genetika. *Jurnal Komtika (Komputasi dan Informatika), 5(1)*, 60-71.

Ranjan, R., & Daniel, A. (2022). An Optimized Deep ConvNet Sentiment Classification Model with Word Embedding and BiLSTM Technique. *Advances in Distributed Computing and Artificial Intelligence Journal*, 309-329.

Riza, M. A., & Charibaldi, N. (2021). Emotion Detection in Twitter Social Media Using Long Short Term Memory (LSTM) and Fast Text. *International Journal of Artificial Intelligence & Robotics*, 15-26.

Rong, X. (2014). word2vec Parameter Learning Explained. *arXiv preprint arXiv:1411.2738.*, 1-21.

Sangeetha, J., & Kumaran, U. (2022). Using BiLSTM Structure with Cascaded Attention Fusion Model for Sentiment Analysis. *Journal of Scientific & Industrial Research*, 444-449.

Satriaji, W., & Kusumaningrum, R. (2018). Effect of Synthetic Minority Oversampling Technique (SMOTE), Feature Representation, and Classification Algorithm on Imbalanced Sentiment Analysis. *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)* (pp. 1-5). IEEE.

Satrya, W. F., Aprilliyani, R., & Yossy, E. H. (2023). Sentiment analysis of Indonesian police chief using multi-level ensemble model. *Procedia Computer Science, 216*, (pp. 620-629).

Subba, B., & Kumari, S. (2022). A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. *Computational Intelligence 38(2)*, 530-559.

Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C. J. (2019). Evaluating Word Embedding Models: Methods and Experimental Results. *APSIPA transactions on signal and information processing*, 1 of 14.

Widyaningrum, I., & Kamayani, M. (2023). Analisis sentimen opini masyarakat terhadap penggunaan layanan maxim menggunakan algoritma naïve bayes.

Jurnal CoSciTech (Computer Science and Information Technology), 4(3), 651-660. doi:10.37859/coscitech.v4i3.6194

Xiaoyan, L., Raga, R. C., & Xuemei, S. (2022). GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Review. *Journal of Sensors*, 1-12.

Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment Analysis of Comment Texts Based on BiLSTM. *IEEE Access*.

Xu, R., Chen, T., Xia, Y., Lu, Q., Liu, B., & Wang, X. (2015). Word Embedding Composition for Data Imbalances in Sentiment and Emtion Classification. *Cognitive Computation, 7*, 226-240. doi:10.1007/s12559-015-9319-y

Yang, L., Li, Y., Wang, J., & Siherratt, R. S. (2020). Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning. *IEEE Access*, 23526.

Zhao, H., Liu, Z., Yao, X., & Yang, Q. (2021). A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach. *Information Processing and Management*.

PUSTAKA LAPORAN PENELITIAN

-

PUSTAKA ELEKTRONIK

-