JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 9, No. 3, SEPTEMBER, 2024, Pp. 217 – 229
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Pelabelan Sentimen Berbasis *Semi-Supervised Learning* menggunakan Algoritma LSTM dan GRU

Puji Ayuningtyas [(1)*], Siti Khomsah [(2)], Sudianto [(3)]
[1,3] Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom Purwokerto, Banyumas, Indonesia
[2] Sains Data, Fakultas Informatika, Institut Teknologi Telkom Purwokerto, Banyumas, Indonesia
e-mail : {20102122,siti,sudianto}@ittelkom-pwt.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 9 Mei 2024, direvisi 25 Juni 2024, diterima 26 Juni 2024, dan dipublikasikan 25 September 2024.

Abstract

In the sentiment analysis research process, there are problems when still using manual labeling methods by humans (expert annotation), which are related to subjectivity, long time, and expensive costs. Another way is to use computer assistance (machine annotator). However, the use of machine annotators also has the research problem of not being able to detect sarcastic sentences. Thus, the researcher proposed a sentiment labeling method using Semi-Supervised Learning. Semi-supervised learning is a labeling method that combines human labeling techniques (expert annotation) and machine labeling (machine annotation). This research uses machine annotators in the form of Deep Learning algorithms, namely the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms. The word weighting method used in this research is Word2Vec Continuous Bag of Word (CBoW). The results showed that the GRU algorithm tends to have a better accuracy rate than the LSTM algorithm. The average accuracy of the training results of the LSTM and GRU algorithm models is 0.904 and 0.913. In contrast, the average accuracy of labeling by LSTM and GRU is 0.569 and 0.592, respectively.

Keywords: Annotation, Deep Learning, GRU, LSTM, Semi-Supervised Learning, Word2Vec

Abstrak

Dalam proses penelitian analisis sentimen, terdapat permasalahan yaitu ketika masih menggunakan metode pelabelan secara manual oleh manusia (*expert annotation*), yaitu terkait subjektivitas, waktu yang lama dan biaya yang mahal. Cara yang lain adalah dengan menggunakan bantuan komputer (*machine annotator*). Tetapi, penggunaan *machine annotator* juga memiliki permasalahan penelitian yaitu kurang mampu mendeteksi kalimat sarkas. Sehingga, peneliti mengusulkan metode pelabelan sentimen menggunakan *semi-supervised learning*. *Semi-supervised learning* adalah metode pelabelan di mana akan menggabungkan teknik pelabelan manusia (*expert annotation*) dan pelabelan mesin (*machine annotation*). Dalam penelitian ini menggunakan *machine annotator* berupa algoritma *deep learning* yaitu algoritma Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU). Metode pembobotan kata digunakan dalam penelitian ini adalah Word2Vec Continuous Bag of Word (CBoW). Hasil penelitian menunjukkan bahwa algoritma GRU cenderung memiliki tingkat akurasi yang lebih baik daripada algoritma LSTM. *Dataset* yang digunakan sebanyak 43.825 baris data dengan perbandingan pembagian data latih dan data uji sebesar 80:20. Akurasi rata-rata hasil pelatihan model algoritma LSTM dan GRU adalah 0,904 dan 0,913. Sedangkan akurasi rata-rata pelabelan oleh LSTM dan GRU masing-masing adalah sebesar 0,569 dan 0,592.

Kata Kunci: Anotasi, *Deep Learning*, GRU, LSTM, *Semi-Supervised Learning*, Word2Vec

1. PENDAHULUAN

Pada awal abad 21, perkembangan teknologi semakin masif di mana percepatan implementasi teknologi cerdas atau revolusi industri 4.0 sudah diterapkan di seluruh dunia termasuk Indonesia. Perkembangan teknologi informasi dan komunikasi ini juga mempengaruhi perubahan pola pada bidang pembayaran dari pembayaran tunai berubah menjadi non tunai. Di Indonesia, perkembangan alat pembayaran terus mengalami perubahan bentuk, mulai dari uang logam,

uang kertas, hingga mengalami evolusi berupa uang yang ditempatkan dalam media elektronik yang disebut dompet digital (Janah & Setiyawan, 2022).

Bank Indonesia mencatat lebih dari 38 aplikasi dompet digital atau *e-wallet*. Menurut laporan E-Wallet Industry Outlook 2023 dari Insight Asia, dari 1.300 warga perkotaan yang disurvei, 74% di antaranya sudah pernah menggunakan dompet digital. Persebaran popularitas aplikasi *e-wallet* pada survei tersebut adalah pada peringkat pertama aplikasi GoPay dengan pengguna sebanyak 71%, peringkat kedua yaitu aplikasi OVO dengan 70%, dan peringkat ketiga teratas yaitu aplikasi Dana dengan 61% pengguna (Setiyawan et al., 2023).

Perkembangan media teknologi informasi yang masif saat ini selaras dengan peningkatan kuantitas data yang tersedia. Banyaknya data yang tersedia seringkali memiliki kelemahan sehingga membutuhkan berbagai pemrosesan tambahan sebelum dilakukan tahap selanjutnya (Rahma & Suadaa, 2023). Salah satunya kelemahan bahwa data yang tersedia tidak banyak yang memiliki kelas label/kategori. Ketersediaan data yang tidak memiliki label banyak ditemukan pada data teks.

Pada data teks, metode pemberian kelas label dapat dilakukan secara manual, seperti yang dilakukan dalam penelitian Khomsah & Aribowo (2020). Teknik pelabelan opini pada penelitian tersebut menggunakan pelabelan manual oleh manusia. Sedangkan dalam penelitian Zhafira et al. (2021), dijelaskan bahwa pelabelan manual menggunakan *human annotator* memiliki kelemahan terkait subjektivitas. Berdasarkan penelitian tersebut, subjektivitas dapat diminimalisir dengan menambah jumlah *annotator*. Pada penelitian tersebut menggunakan 3 *annotator* yang berasal dari bidang ilmu bahasa, ilmu psikologi, dan teknik komputer. Hal tersebut membutuhkan waktu yang lama dan biaya yang mahal.

Cara yang lain adalah dengan menggunakan bantuan komputer (*machine annotator*). Dalam penelitian Bandhakavi et al. (2017) yang menggunakan metode *sentiment lexicons* dan General-Purpose Emotion Lexicons (GPELs) seperti WordNet-Affect2 sebagai *machine annotator*, dijelaskan bahwa *machine annotator* memiliki permasalahan di mana dalam media sosial (misal: Twitter) manusia lebih memilih menggunakan kosakata informal dan emoji untuk menyampaikan emosi, daripada menggunakan kosakata formal seperti pada GPEL. Selain itu, hubungan antara kata-kata dan emosi bervariasi dari satu domain ke domain lain, ini lebih dikenal dengan disambiguasi kontekstual.

Berdasarkan permasalahan yang telah dijabarkan, diperlukan sebuah solusi di mana anotasi dapat dilakukan otomatis sehingga dapat meminimalisir terjadinya disambiguasi kontekstual, sekaligus dapat mengurangi durasi dalam proses anotasi. Maka, penelitian ini bertujuan mengusulkan pemodelan anotasi (pelabelan) dengan "Pelabelan Sentimen Berbasis Semi-Supervised Learning menggunakan Deep Learning."

Semi-Supervised Learning (SSL) merupakan gabungan antara anotasi menggunakan *human annotator* dan *machine annotator.* SSL pernah dilakukan penelitian, misalnya pada penelitian Wisnalmawati et al. (2022) menggunakan metode pelabelan manual, dengan pakar sebagai manusia yang menentukan label dalam korpus. Tetapi, untuk membuat korpus berlabel lengkap dengan kualitas tinggi memerlukan banyak usaha, waktu, dan biaya, serta dapat menjadi tugas yang berat. Tujuan dilakukan penelitian ini adalah untuk mengetahui bagaimana kombinasi antara Term Frequency–Inverse Document Frequency (TF-IDF) dengan Random Forest (RF) untuk meningkatkan akurasi ketika digunakan pada SSL. Penelitian ini menjabarkan hasil penelitian bahwa pada Data1 dengan jumlah kelas data sebanyak 3, Random Forest memiliki F1-*score* 0,65 sedangkan Naive Bayes (NB) 0,62. Hal ini menunjukkan bahwa RF bekerja lebih baik daripada NB pada data 3 kelas. Sedangkan pada Data2 yang memiliki 2 kelas, NB memiliki F1-*score* 0,76 sedangkan RF 0,71.

Selanjutnya pada penelitian Anggraini et al. (2021) bertujuan untuk menerapkan metode untuk memberikan pelabelan sentimen berdasarkan kata kunci dengan tetap memperhatikan konteks

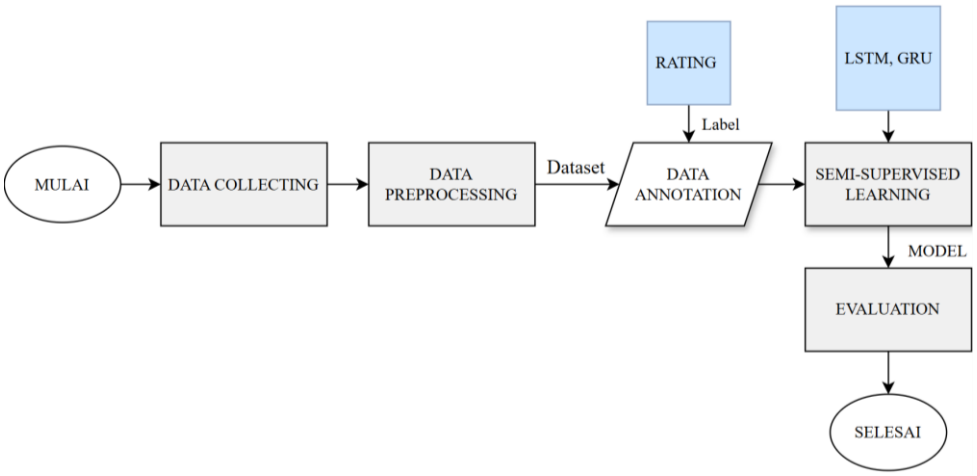
kalimat. Penelitian ini dilatarbelakangi oleh penggunaan metode Lexicon dalam pelabelan *dataset* hanya memberikan label berdasarkan makna setiap kata penyusunnya. Sehingga tidak dapat mengetahui arah konteks komentar, dan kurang dapat efektif untuk mengatasi kalimat sarkas. Metode penelitian yang diusulkan oleh peneliti menggunakan metode TF-IDF untuk *word embedding*, dan algoritma Latent Semantic Index (LSI)/Latent Semantic Analysis (LSA) untuk pemaknaan kata terhadap konteks kalimat. Pada hasil penelitian mengkombinasikan metode menggunakan TF-IDF dan LSA mampu menemukan detail permasalahan. Contohnya kata "vaksin" di TF-IDF menempati rangking pertama positif, negatif, maupun netral dengan masing-masing memiliki bobot 0,69, 0,77, dan 0,69.

Dalam penelitian ini menggunakan *machine annotator* berupa algoritma *deep learning* yaitu algoritma Long Short-Term Memory (LSTM) dan algoritma Gated Recurrent Unit (GRU). LSTM dan GRU merupakan algoritma turunan dari Recurrent Neural Network (RNN) yang banyak digunakan dalam penelitian penambangan data teks. Performa Algoritma LSTM dijelaskan melalui penelitian (Seabe et al., 2023), yang bertujuan membandingkan performa algoritma LSTM dengan algoritma algoritma Gated Reccurent Unit (GRU) yang diterapkan pada permasalahan *forecasting*. Hasil penelitian didapatkan bahwa algoritma LSTM memiliki performa yang lebih baik daripada GRU dengan hasil RMSE LSTM sebesar 0,039 dan MAPE 1031,34. Sedangkan algoritma GRU memiliki nilai RMSE sebesar 0,057 dan MAPE 1274,17.

Alasan digunakannya algoritma LSTM adalah kemampuan algoritma LSTM dalam memproses data sekuensial yang panjang sehingga mampu mengatasi permasalahan *vanishing gradient* (Oktaviani & Hustinawati, 2021), dan memiliki tingkat akurasi klasifikasi cenderung lebih baik (Ezen-Can, 2020; Romadhoni & Holle, 2022). Algoritma GRU merupakan algoritma *deep learning* yang memiliki struktur model mirip dengan Algoritma LSTM dengan versi yang lebih sederhana sehingga memiliki waktu komputasi yang labih singkat. Misalnya, pada penelitian Nosouhian et al. (2021) dengan membandingkan algoritma GRU dan LSTM dengan waktu komputasi GRU yang lebih singkat daripada LSTM pada seluruh skenario.

2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini ditunjukkan pada Gambar 1. Proses dimulai dengan pengumpulan data, diikuti oleh pra pemrosesan data untuk membersihkan dan menyiapkan data sebelum analisis lebih lanjut. Setelah itu, dilakukan data annotation untuk memberikan label pada data yang diperlukan. Selanjutnya, dataset dibagi menjadi beberapa bagian untuk pelatihan dan pengujian model. Proses *semi-supervised learning* diterapkan untuk memanfaatkan data berlabel dan tidak berlabel guna meningkatkan akurasi model. Akhirnya, evaluasi model dilakukan untuk menilai kinerja dan efektivitas dari algoritma yang digunakan. Metodologi ini memastikan bahwa setiap tahap dilakukan secara sistematis untuk mencapai hasil yang optimal dalam penelitian.



Gambar 1 *Flowchart* Penelitian

2.1 Pengumpulan Data (*Data Colecting*)

Data dikumpulkan melalui proses scraping aplikasi yang terdapat pada Google Play Store. Proses ini menggunakan *library* yang tersedia pada bahasa pemrograman Python, yaitu google-play-scraper. *Dataset* yang digunakan ada 3 macam, yaitu *dataset* komentar *review e-wallet* Dana, Ovo, dan GoPay dari Google PlayStore. *Dataset* yang digunakan sebanyak 43.825 baris data dengan rincian *dataset* pertama, menggunakan data yang diambil dari PlayStore pada komentar aplikasi Dana sebanyak 18.353 data. *Dataset* kedua diambil dari PlayStore pada komentar *review e-wallet* Ovo sebanyak 23.880 data. *Dataset* ketiga diambil dari PlayStore pada komentar aplikasi GoPay sebanyak 1.592 data.

2.2 Pra Pemrosesan Data (*Data Preprocessing*)

Pra pemrosesan data sangat penting dilakukan karena data yang diperoleh sering kali masih mengandung *noise*, seperti data *null*, data duplikat, dan berbagai ketidaksesuaian lainnya. Oleh karena itu, proses pembersihan data perlu dilakukan untuk memastikan kualitas dan keakuratan data yang akan digunakan dalam analisis. Dalam penelitian ini, data *preprocessing* terdiri dari enam tahap, yaitu:

2.2.1 *Drop null* dan *drop duplicate*

Null dapat memengaruhi kinerja algoritma dalam melakukan proses selanjutnya. Sehingga adanya nilai kosong (*null*) pada suatu *dataset* harus dilakukan penanganan. Terdapat 2 (dua) cara yang dapat dilakukan, yaitu dengan menghapus *null*, dan mengisi *null* dengan nilai tertentu (Khatri & P, 2020). *Dataset* OVO terdapat 217 data duplikat, dan tidak memiliki data kosong (null). *Dataset* DANA terdapat 16 data duplikat, dan tidak memiliki data kosong (null). Sedangkan *dataset* GOPAY tidak terdapat data duplikat maupun data kosong, sehingga jumlah baris data tidak berkurang setelah melewati proses ini.

2.2.2 *Cleaning data*

Cleaning data merupakan proses penting untuk membersihkan data dari komponen tertentu yang tidak diperlukan, seperti URL, *username*, dan *hashtags*, guna memastikan kualitas dan relevansi informasi yang akan dianalisis (Arsi & Waluyo, 2021). Proses ini membantu menghilangkan elemen yang dapat menyebabkan kebisingan dalam dataset, sehingga analisis yang dilakukan menjadi lebih akurat. Hasil dari cleaning data dapat dilihat pada Tabel 1.

Tabel 1 Hasil *Cleaning Data*

Sebelum *Cleaning*	Setelah *Cleaning*
Woi aplikasi gak jelas lu , susah amat dibuka, gimana mau chat	Woi aplikasi gak jelas lu susah amat dibuka gimana mau chat

2.2.3 *Case folding*

Case folding adalah proses *preprocessing* data teks yang bertujuan untuk mengubah seluruh huruf dalam *dataset* menjadi huruf kecil atau *lowercase*, sehingga mengurangi variasi data yang disebabkan oleh perbedaan penggunaan huruf besar dan kecil (Af'idah et al., 2021; Ayuningtyas & Tantyoko, 2024). Proses ini penting dalam analisis teks karena membantu menyederhanakan data dan memastikan bahwa istilah yang sama tidak diperlakukan sebagai entitas yang berbeda hanya karena perbedaan kapitalisasi. Hasil dari proses *case folding* dapat dilihat pada Tabel 2, yang menunjukkan perbedaan antara data sebelum dan setelah proses ini dilakukan.

Tabel 2 Hasil *Case Folding*

Sebelum *Case Folding*	Setelah *Case Folding*
Woi aplikasi gak jelas lu susah amat dibuka gimana mau chat	woi aplikasi gak jelas lu susah amat dibuka gimana mau chat

2.2.4 *Tokenizing*

Merupakan tahap pemisahan kalimat dalam *dataset* berdasarkan tiap kata penyusunnya. Kata yang telah dipisah dari rangkaian kalimat disebut *token* atau *term* (Ayuningtyas & Tantyoko, 2024). *Term* ini akan diberikan bobot kata pada tahap pembobotan kata (Romadhoni & Holle, 2022). Hasil *tokenizing* dapat dilihat pada Tabel 3, yang menunjukkan perbedaan sebelum dan setelah proses tokenisasi.

Tabel 3 Hasil Tokenisasi

Sebelum *Tokenizing*	Setelah *Tokenizing*
woi aplikasi gak jelas lu susah amat dibuka gimana mau chat	['woi', 'aplikasi', 'gak', 'jelas', 'lu', 'susah', 'amat', 'dibuka', 'gimana', 'mau', 'chat']

2.2.5 *Slang word removal*

Penghapusan kata *slang* (*slang word removal*) bertujuan untuk mengubah kata-kata tidak baku menjadi kata baku, sehingga meningkatkan kualitas dan formalitas data yang akan dianalisis (Gifari et al., 2022). Proses ini sangat penting dalam konteks analisis teks, terutama ketika data berasal dari sumber yang tidak terstruktur, seperti media sosial atau forum *online*. Hasil dari proses *slang word removal* dapat dilihat pada Tabel 4.

Tabel 4 Hasil *Slang Word Removal*

Sebelum *Slang Word Removal*	Setelah *Slang Word Removal*
['woi', 'aplikasi', 'gak', 'jelas', 'lu', 'susah', 'amat', 'dibuka', 'gimana', 'mau', 'chat']	['', 'aplikasi', 'tidak', 'jelas', 'kamu', 'susah', 'amat', 'dibuka', 'bagaimana', 'mau', 'pesan']

2.2.6 *Under sampling*

Under sampling merupakan salah satu metode yang dapat digunakan untuk mengatasi ketidakseimbangan kuantitas data antarkelas. Cara kerja *under sampling* adalah dengan membuang sampel (secara acak) dari kelas mayoritas, sehingga kuantitas dari kelas minoritas dan mayoritas akan sama (Magnolia et al., 2022). Hasil dari proses *under sampling*, diperoleh banyak *dataset* Ovo pada *rating* 1, 2, 4, dan 5 masing-masing 4.943 data, dan banyak *dataset* Ovo pada *rating* 1, 2, 4, dan 5 masing-masing 4.000 data.

2.3 *Data Annotation*

Setelah dilakukan *under sampling*, masuk pada tahap pengelompokan kategori kelas berdasarkan *rating* komentar, yaitu *rating* 1 dan 2 termasuk pada kategori negatif, *rating* 4 dan 5 termasuk kategori positif. Hasil yang diperoleh pada tahap pelabelan ini adalah *dataset* Ovo memiliki banyak data pada kelas positif dan negatif masing-masing sebanyak 9.886 data. Sedangkan *dataset* Dana memiliki banyak data pada kelas positif dan negatif masing-masing sebanyak 8.000 data.

2.4 Pembagian *Dataset*

Setelah dilakukan pengelompokan kelas menjadi kelas positif dan negatif, tahap selanjutnya adalah menggabungkan *dataset* Ovo dan Dana menjadi satu *dataset* gabungan. Penggabungan ini bertujuan untuk menyatukan data dari kedua *platform e-wallet* sehingga analisis dapat dilakukan secara lebih komprehensif. *Dataset* gabungan ini telah melalui proses *preprocessing*, seperti pembersihan data dan pengelompokan kategori kelas. Contoh *dataset* yang telah diproses dan dikelompokkan ke dalam kategori kelas positif dan negatif dapat dilihat pada Tabel 5.

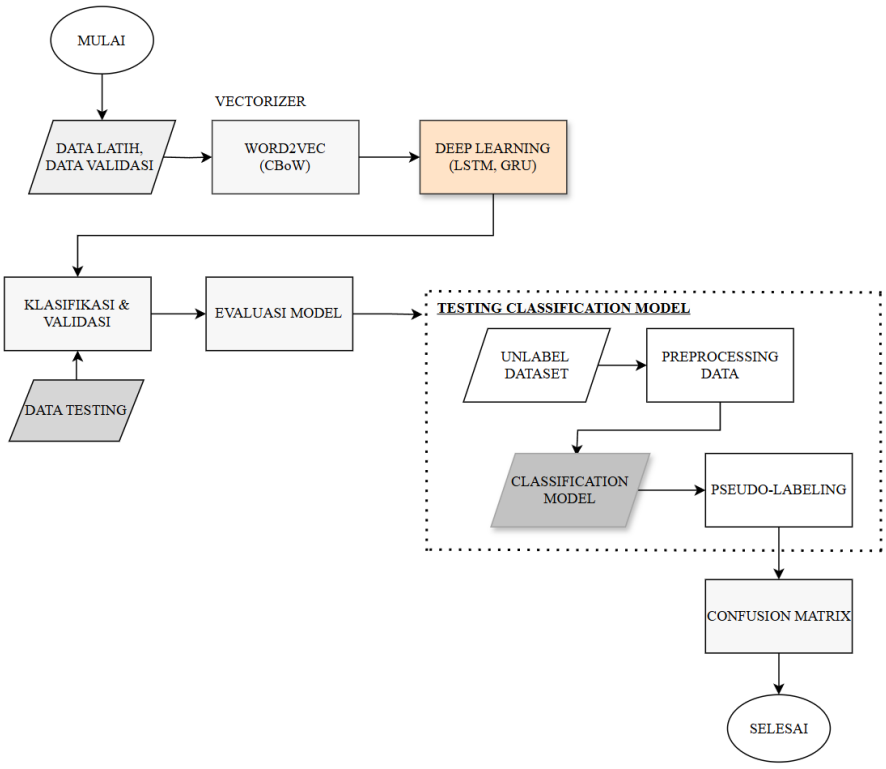
Pembagian *dataset* (*splitting data*) dilakukan setelah proses pelabelan data, yaitu dengan membagi *dataset* yang digunakan dengan perbandingan yang telah ditentukan. *Splitting data*

dengan membagi data sebanyak 35.772 ke dalam 3 (tiga) kategori, yaitu data latih, data validasi, dan data uji. Langkah pertama adalah mengambil 386 data pada masing-masing kelas untuk menjadi data uji, atau total sebanyak 772 data. Kemudian sebanyak 35.000 data dibagi menjadi data latih dan data validasi dengan perbandingan 80:20. Membagi data menjadi 80% untuk pelatihan dan 20% untuk pengujian memberikan cukup data untuk melatih model sambil menyisakan cukup data untuk menguji kinerjanya secara independen. Ini berguna dalam menentukan perbandingan yang efisien secara praktis, untuk mengantisipasi model mengalami *overfitting* (terlalu cocok dengan data latih) dan juga *underfitting* (tidak cukup belajar dari data latih). Sebelum dilakukan pembagian ke data latih dan data validasi, data dilakukan cek terhadap data duplikat dan data *null*, serta *under sampling*. Diperoleh banyak data yang akan dibagi ke data latih dan data validasi sebanyak 34.884 data. Dengan data tersebut, didapatkan data latih sebanyak 27.907 data, dan data validasi sebanyak 6.977 data.

Tabel 5 Contoh *Dataset*

No.	Data	*Score*
1	pelayanan buruk yang pernah saya temui di aplikasi transaksi uang masak saya transfer dana dari jam pagi jam sore tidak masuk pas ke lagi di proses terus tidak sudah aplikasi macam apa ini tolong main toko ditegur lah apa di baned sekali bukan dikit saya melakukan transaksi uang padahal itu kan uang saya sendiri dan transfer juga ke rekening saya sendiri	0
2	halo min biasanya transaksi dengan ovo sangat aman dan mudah tapi semalam mungkin ada gangguan jaringan dan saya ada coba isi pulsa tapi kok sampai sekarang belum masuk pulsanya tapi saldonya kepotong saya harus bagaimana iya mohon solusinya terima kasih	0
3	terima kasih buat aplikasi ovo tinkatkan layanan untuk kedepan yang lebih baik	1
4	akhir ini server sering error padahl sinyal wifi penuh isi paket data pun sering gagal	1

2.5 *Semi-Supervised Learning*



Gambar 2 Alur Metode *Semi-Supervised Learning*

Semi-Supervised Learning (SSL) merupakan *framework* untuk memberikan label pada sejumlah besar data yang tidak berlabel (Zhou, 2021). Teknik SSL dapat meningkatkan kinerja model pada tugas *machine learning*, misalnya klasifikasi teks, terjemahan mesin, klasifikasi gambar. Cara kerja SSL adalah dengan mengambil *dataset* berlabel yang sudah ada dan hanya menggunakan sebagian kecil data pelatihan sebagai data berlabel, sementara memperlakukan sisa data sebagai *dataset* tidak berlabel (Ouali et al., 2020). Alur metode *semi-supervised learning* pada penelitian ini terdapat pada Gambar 2.

Tahap *semi-supervised learning* dimulai dengan menentukan *dataset* yang akan digunakan sebagai masukan (input). *Dataset* yang telah melewati tahap *preprocessing* dan tahap *binary labeling*, kemudian dibagi ke dalam data latih, data validasi, dan data uji. Ketiga kategori data tersebut, masing-masing masuk dalam tahap vektorisasi yang mengubah kata menjadi bentuk vektor numerik. Tahap ini menggunakan metode Word2Vec, dengan model Continuous Bag of Word (CBoW). Model CBoW dilatih dengan 120 *epoch* dengan *hyperparameter* vector_size=100, window=5, min_count=1, sg=0. Lebih lanjut skenario pemodelan algoritma berdasarkan *hyperparameter* yang sudah ditentukan, yaitu untuk skenario pemodelan algoritma LSTM dapat dilihat pada Tabel 6, dan skenario pemodelan algoritma GRU dapat dilihat pada Tabel 7.

Tabel 6 Skenario Pemodelan Algoritma LSTM

Learning Rate (LR)	Koefisien Regularisasi (l2)	*Epoch*	*Batch Size*	Skenario
0.002	0.001 dan 0.001	50	128	*0.002-LSTM-50-128*
			256	*0.002-LSTM-50-256*
			512	*0.002-LSTM-50-512*
		100	128	*0.002-LSTM-100-128*
			256	*0.002-LSTM-100-256*
			512	*0.002-LSTM-100-512*
0.001	0.005 dan 0.01	50	128	*0.001-LSTM-50-128*
			256	*0.001-LSTM-50-256*
			512	*0.001-LSTM-50-512*
		100	128	*0.001-LSTM-100-128*
			256	*0.001-LSTM-100-256*
			512	*0.001-LSTM-100-512*

Tabel 7 Skenario Pemodelan Algoritma GRU

Learning Rate (LR)	Koefisien Regularisasi (l2)	*Epoch*	*Batch Size*	Skenario
0.002	0.001 dan 0.001	50	128	*0.002-GRU-50-128*
			256	*0.002-GRU-50-256*
			512	*0.002-GRU-50-512*
		100	128	*0.002-GRU-100-128*
			256	*0.002-GRU-100-256*
			512	*0.002-GRU-100-512*
0.001	0.005 dan 0.01	50	128	*0.001-GRU-50-128*
			256	*0.001-GRU-50-256*
			512	*0.001-GRU-50-512*
		100	128	*0.001-GRU-100-128*
			256	*0.001-GRU-100-256*
			512	*0.001-GRU-100-512*

Mencoba berbagai skenario memungkinkan evaluasi kinerja model di bawah kondisi yang berbeda, membantu menemukan konfigurasi yang menghasilkan akurasi dan efisiensi terbaik. Eksperimen dengan skenario yang berbeda memungkinkan peneliti untuk mengoptimalkan model secara menyeluruh, termasuk mengurangi *overfitting* atau *underfitting*. Dengan

memvariasikan *learning rate*, koefisien regularisasi, *epoch*, dan *batch size*, penelitian dapat mengidentifikasi kombinasi parameter yang optimal untuk model LSTM dan GRU.

2.6 Evaluasi Model

Confusion matrix merupakan salah satu metode yang umum digunakan dalam metode evaluasi algoritma yang melakukan komputasi data berlabel (*supervised learning*). Cara kerja *confusion matrix* adalah dengan mengolah data dengan tujuan membandingkan hasil prediksi dengan data sesungguhnya. Terdapat empat macam bagian evaluasi dari *confusion matrix*, yaitu akurasi, *recall, precision,* dan F1 *score*. *Confusion matrix* dapat dilihat pada Tabel 8.

Tabel 8 *Confusion Matrix*

Aktual	Prediksi	
	Positif	Negatif
Positif	True Positive (TP)	False Negative (FN)
Negatif	False Positive (FP)	True Negative (TN)

1) Akurasi, yaitu ukuran tingkat kedekatan antara nilai sebenarnya dengan nilai hasil prediksi model (Rolangon et al., 2023). Hasil model dapat dikatakan semakin baik apabila menghasilkan nilai akurasi yang mendekati 100%. Rumus akurasi dituliskan pada Pers. (1).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \times 100\% \quad (1)$$

2) *Recall*, atau disebut dengan sebut *sensitivity* merupakan metode yang digunakan untuk mengevaluasi seberapa baik model mengelompokkan data berlabel positif secara keseluruhan (Rolangon et al., 2023). Rumus *recall* dituliskan pada Pers. (2).

$$Recall = \frac{TP}{(TP + FN)} \times 100\% \quad (2)$$

3) *Precision*, digunakan untuk mengukur tingkat ketepatan suatu model yang dibangun dalam mengklasifikasikan suatu data kedalam kelas positif (Suryati et al., 2023). Rumus *precision* dituliskan pada Pers. (3).

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

4) F1 *score*, adalah perbandingan rata-rata *recall* dan presisi (*precision*) (Ayuningtyas & Tantyoko, 2024), dengan rumus seperti pada Pers. (4).

$$F1\ score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \times 100\% \quad (4)$$

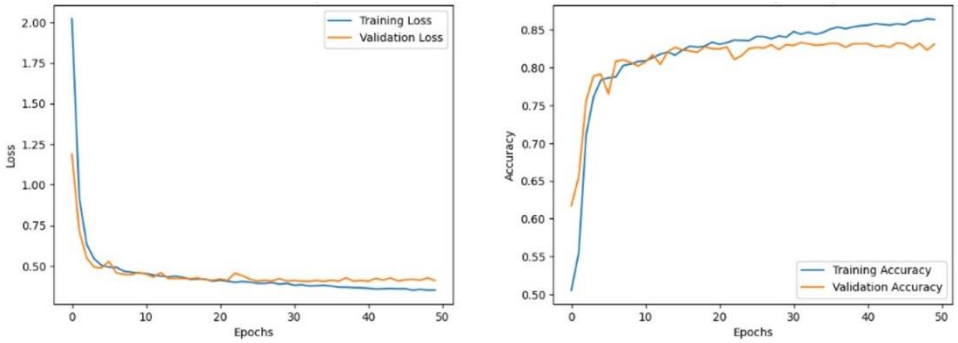
3. HASIL DAN PEMBAHASAN

3.1 Pemodelan Algoritma *Deep Learning*

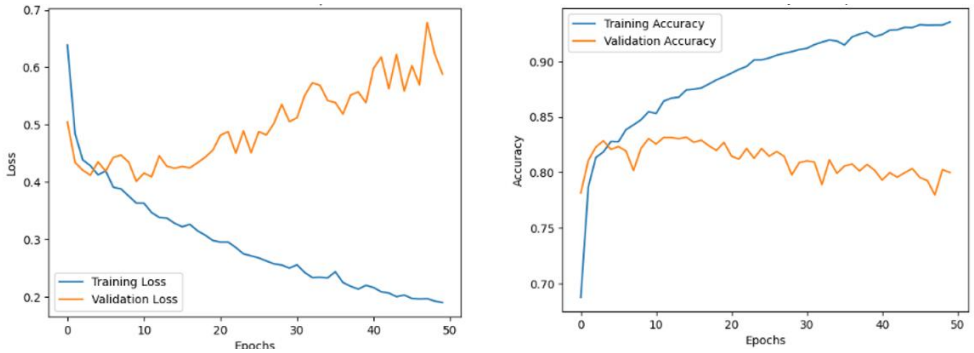
3.1.1 Hasil Pelatihan Algoritma LSTM

Pada skenario *0.001-LSTM-50-512,* model dilatih menggunakan Algoritma LSTM dengan *hyperparameter* Learning Rate Optimasi ADAM sebesar 0,001, *epoch* 50, Koefisien Regularisasi (l2) *layer* pertama dan kedua masing-masing 0,005 dan 0,01, dan *batch size* 512. Algoritma LSTM dapat mencapai kondisi *good fit* pada susunan hyperparameter ini. Sedangkan pada skenario *0.002-LSTM-50-512,* dengan *hyperparameter* Learning Rate Optimasi ADAM sebesar 0,002,

epoch 50, Koefisien Regularisasi (l2) *layer* pertama dan kedua masing-masing 0,001 dan 0,001, dan *batch size* 512. Algoritma LSTM berada pada kondisi *overfit.* Perbandingan performa kedua skenario LSTM dapat dilihat pada Gambar 3 dan Gambar 4.



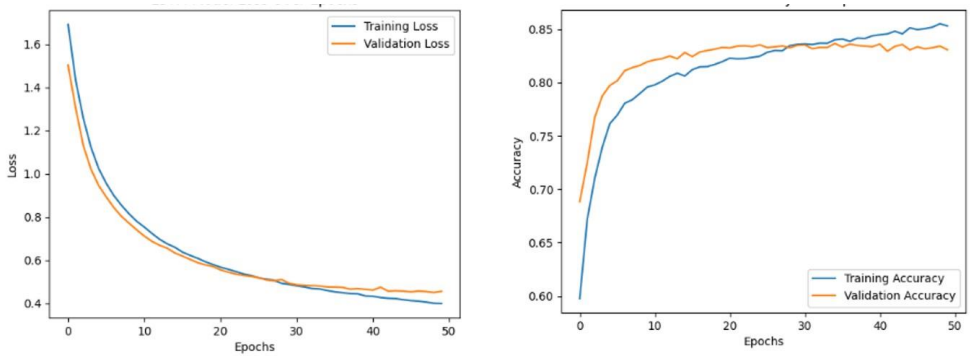
Gambar 3 Performa Skema *0.001-LSTM-50-512*



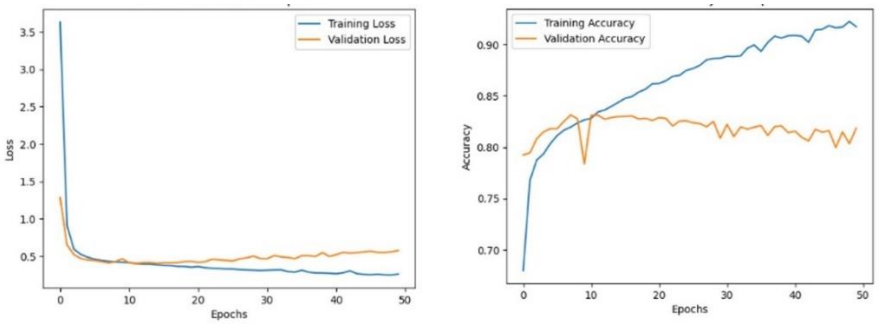
Gambar 4 Performa Skema *0.002-LSTM-50-512*

3.1.2 Hasil Pelatihan Algoritma GRU

Pada skenario *0.002-GRU-50-128,* model dilatih menggunakan Algoritma GRU dengan *hyperparameter* Learning Rate (LR) Optimasi ADAM sebesar 0,002, *epoch* 50, Koefisien Regularisasi (l2) *layer* pertama dan kedua masing-masing 0,001, dan *batch size* 128. Algoritma GRU dapat mencapai kondisi *good fit* pada susunan *hyperparameter* ini. Sedangkan pada skenario *0.001-GRU-50-128,* dengan *hyperparameter* Learning Rate Optimasi ADAM sebesar 0,001, *epoch* 50, Koefisien Regularisasi (l2) *layer* pertama dan kedua masing-masing 0,005 dan 0,01, dan *batch size* 128. Algoritma GRU berada pada kondisi *overfit.* Perbandingan performa kedua skenario GRU dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5 Performa Skema *0.002-GRU-50-128*



Gambar 6 Performa Skema *0.001-GRU-50-128*

3.1.3 Evaluasi Pelatihan Algoritma *Deep Learning*

Hasil evaluasi model algoritma LSTM menggunakan data uji terdapat pada Tabel 9. Sedangkan, hasil evaluasi model algoritma GRU menggunakan data uji dapat dilihat pada Tabel 10. Dari kedua tabel tesebut, dapat diketahui bahwa algoritma LSTM memiliki nilai akurasi, presisi, *recall*, dan F1-*score* tertinggi masing-masing sebesar 0,94, 0,966, 0,943, dan 0,939. Sedangkan GRU memiliki nilai akurasi, presisi, *recall*, dan F1-*score* tertinggi masing-masing sebesar 0,952, 0,951, 0,961, dan 0,952. Dengan demikian, pada penelitian ini, algoritma GRU memiliki kemampuan klasifikasi yang lebih baik daripada LSTM.

Tabel 9 Hasil Evaluasi Algoritma LSTM

Skenario LSTM	Akurasi	Presisi	*Recall*	F1-*Score*
0.002-LSTM-50-128	0,922	0,95	0,891	0,92
0.002-LSTM-50-256	0,885	0,845	0,943	0,891
0.002-LSTM-50-512	0,903	0,959	0,842	0,897
0.002-LSTM-100-128	0,929	0,949	0,907	0,927
0.002-LSTM-100-256	0,94	0,955	0,925	0,939
0.002-LSTM-100-512	0,938	0,947	0,927	0,937
0.001-LSTM-50-128	0,891	0,942	0,834	0,885
0.001-LSTM-50-256	0,883	0,911	0,85	0,879
0.001-LSTM-50-512	0,876	0,818	0,966	0,886
0.001-LSTM-100-128	0,896	0,875	0,925	0,899
0.001-LSTM-100-256	0,895	0,932	0,852	0,89
0.001-LSTM-100-512	0,889	0,936	0,834	0,882

Tabel 10 Hasil Evaluasi Algoritma GRU

Skenario GRU	Akurasi	Presisi	*Recall*	F1-*Score*
0.002-GRU-50-128	0,872	0,877	0,865	0,871
0.002-GRU-50-256	0,931	0,924	0,940	0,932
0.002-GRU-50-512	0,946	0,932	0,961	0,946
0.002-GRU-100-128	0,900	0,889	0,915	0,902
0.002-GRU-100-256	0,887	0,919	0,850	0,883
0.002-GRU-100-512	0,873	0,873	0,873	0,873
0.001-GRU-50-128	0,921	0,945	0,894	0,919
0.001-GRU-50-256	0,902	0,928	0,870	0,898
0.001-GRU-50-512	0,891	0,913	0,865	0,888
0.001-GRU-100-128	0,942	0,945	0,938	0,941
0.001-GRU-100-256	0,942	0,945	0,938	0,941
0.001-GRU-100-512	0,952	0,951	0,953	0,952

3.2 Pengujian Model *Semi-Supervised Learning*

Setelah data dibersihkan melalui tahap *preprocessing*, langkah berikutnya adalah melakukan pengujian model *supervised learning* menggunakan *dataset* tanpa label (*unlabeled dataset*). Pengujian ini bertujuan untuk mengevaluasi performa model *deep learning* dalam menangani data yang belum diberi label. Hasil pengujian model Long Short-Term Memory (LSTM) terhadap dataset tanpa label dapat dilihat pada Tabel 11, yang memperlihatkan kinerja model dalam mengenali pola dari data. Sementara itu, hasil pengujian model Gated Recurrent Unit (GRU) terhadap *dataset* tanpa label disajikan pada Tabel 12. Kedua tabel ini menampilkan hasil perbandingan akurasi dan performa masing-masing model dalam mengolah *dataset* yang sama, sehingga dapat diidentifikasi model mana yang lebih efektif dalam skenario ini.

Tabel 11 Tabel Hasil Percobaan Pelabelan (SSL) Algoritma LSTM

Skenario LSTM	Akurasi	Presisi	*Recall*	F1-*Score*
0.002-LSTM-50-128	0,538	0,537	0,554	0,545
0.002-LSTM-50-256	0,566	0,602	0,388	0,472
0.002-LSTM-50-512	0,568	0,556	0,668	0,607
0.002-LSTM-100-128	0,560	0,559	0,569	0,564
0.002-LSTM-100-256	0,571	0,566	0,607	0,586
0.002-LSTM-100-512	0,553	0,556	0,556	0,544
0.001-LSTM-50-128	0,588	0,580	0,641	0,609
0.001-LSTM-50-256	0,602	0,603	0,595	0,599
0.001-LSTM-50-512	0,575	0,559	0,707	0,625
0.001-LSTM-100-128	0,546	0,568	0,389	0,462
0.001-LSTM-100-256	0,581	0,588	0,540	0,563
0.001-LSTM-100-512	0,579	0,568	0,653	0,608

Tabel 12 Tabel Hasil Percobaan Pelabelan (SSL) Algoritma GRU

Skenario GRU	Akurasi	Presisi	*Recall*	F1-*Score*
0.002-GRU-50-128	0,638	0,653	0,588	0,619
0.002-GRU-50-256	0,583	0,592	0,533	0,561
0.002-GRU-50-512	0,584	0,586	0,570	0,578
0.002-GRU-100-128	0,599	0,622	0,608	0,559
0.002-GRU-100-256	0,606	0,602	0,624	0,613
0.002-GRU-100-512	0,601	0,606	0,573	0,589
0.001-GRU-50-128	0,572	0,567	0,609	0,587
0.001-GRU-50-256	0,573	0,571	0,590	0,581
0.001-GRU-50-512	0,607	0,605	0,617	0,611
0.001-GRU-100-128	0,574	0,575	0,570	0,573
0.001-GRU-100-256	0,594	0,595	0,585	0,590
0.001-GRU-100-512	0,574	0,580	0,535	0,557

Dari Tabel 11 dan Tabel 12, dapat diketahui bahwa algoritma LSTM memiliki nilai akurasi, presisi, *recall*, dan F1-*score* tertinggi masing-masing sebesar 0,602, 0,603, 0,707, dan 0,625. Sedangkan GRU memiliki nilai akurasi, presisi, *recall*, dan F1-*score* tertinggi masing-masing sebesar 0,638, 0,653, 0,624, dan 0,619. Hasil pelatihan dan pengujian model bergantung pada kualitas dan kuantitas data yang digunakan. Pada penelitian ini, akurasi pengujian model SSL masih kurang optimal. Hal ini disebabkan data yang digunakan masih terdapat beberapa kelemahan. Salah satunya tidak konsisten antara kecenderungan komentar yang dituliskan oleh pengguna dan *rating* yang diberikan. Contohnya pada Tabel 5 nomor 4, di mana kecenderungan komentar mengarah ke kategori negatif, namun oleh pengguna diberikan *rating* positif. Ini menyebabkan kebingungan model dalam mempelajari suatu pola klasifikasi, sehingga model memiliki akurasi yang rendah ketika melakukan pelabelan terhadap data baru tanpa label.

4. KESIMPULAN

Algoritma LSTM mencapai *goodfit* ketika menggunakan skema *0.001-LSTM-50-512*, dan mengalami *overfitting* pada skema *0.002-LSTM-50-512.* Parameter *learning rate* pada LSTM dengan nilai 0,001 memiliki hasil evaluasi model yang lebih baik daripada nilai 0,002. Sedangkan Algoritma GRU mencapai *goodfit* ketika menggunakan skema *0.002-GRU-50-128* dan mengalami *overfitting* pada skema *0.001-GRU-50-128.* Parameter *learning rate* pada GRU dengan nilai 0,002 memiliki hasil evaluasi model yang lebih baik daripada nilai 0,001. Algoritma LSTM dan GRU memiliki titik *goodfit* dan *overfit* masing-masing yang dapat memengaruhi hasil evaluasi model, baik ketika pelatihan maupun pengujian model. Hasil pelatihan dan pengujian model bergantung pada kualitas dan kuantitas data yang digunakan.

Penelitian selanjutnya diharap mengoptimalkan pembersihan data, *preprocessing data*, dan melakukan cek ulang terkait kesesuaian komentar dan label sehingga dapat memberikan hasil akurasi pelabelan yang lebih optimal. Menggunakan nilai *hyperparameter* yang lain, misal menambahkan jumlah epoch, mengubah fungsi aktivasi algoritma, menaikkan nilai *learning rate*, dan yang lainnya. Menggunakan metode *word embedding* lain, misal FastText, Glove, atau BERT.

DAFTAR PUSTAKA

Af'idah, D. I., Dairoh, D., Handayani, S. F., & Pratiwi, R. W. (2021). Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen. *Jurnal Informatika: Jurnal Pengembangan IT*, *6*(3), 156–161. https://doi.org/10.30591/jpit.v6i3.3016

Anggraini, N., Harahap, E. S. N., & Kurniawan, T. B. (2021). Text Mining - Analisis Teks Terkait Isu Vaksinasi COVID-19 (Text Mining - Text Analysis Related to COVID-19 Vaccination Issues). *JURNAL IPTEKKOM Jurnal Ilmu Pengetahuan & Teknologi Informasi*, *23*(2), 141–153. https://doi.org/10.17933/iptekkom.23.2.2021.141-153

Arsi, P., & Waluyo, R. (2021). Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *8*(1), 147. https://doi.org/10.25126/jtiik.0813944

Ayuningtyas, P., & Tantyoko, H. (2024). Comparison of the Word2vec Skipgram Model Method Linkaja Application Review using Bidirectional LSTM Algorithm and Support Vector Machine. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, *12*(1), 189. https://doi.org/10.26418/justin.v12i1.72530

Bandhakavi, A., Wiratunga, N., Massie, S., & Padmanabhan, D. (2017). Lexicon Generation for Emotion Detection from Text. *IEEE Intelligent Systems*, *32*(1), 102–108. https://doi.org/10.1109/MIS.2017.22

Ezen-Can, A. (2020). *A Comparison of LSTM and BERT for Small Corpus*. http://arxiv.org/abs/2009.05451

Gifari, O. I., Adha, Muh., Freddy, F., & Durrand, F. F. S. (2022). Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine. *Journal of Information Technology*, *2*(1), 36–40. https://doi.org/10.46229/jifotech.v2i1.330

Janah, L. N., & Setiyawan, S. (2022). Dampak Pandemi Covid-19 Terhadap Penggunaan Dompet Digital Di Indonesia. *Journal of Educational and Language Research*, *1*(7), 709–716. https://doi.org/https://doi.org/10.53625/joel.v1i7.1463

Khatri, A., & P, P. (2020). Sarcasm Detection in Tweets with BERT and GloVe Embeddings. *Proceedings of the Second Workshop on Figurative Language Processing*, 56–60. https://doi.org/10.18653/v1/2020.figlang-1.7

Khomsah, S., & Aribowo, A. S. (2020). Text-Preprocessing Model Youtube Comments in Indonesian. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *4*(4), 648–654. https://doi.org/10.29207/resti.v4i4.2035

Magnolia, C., Nurhopipah, A., & Kusuma, B. A. (2022). Penanganan Imbalanced Dataset untuk Klasifikasi Komentar Program Kampus Merdeka Pada Aplikasi Twitter. *Edu Komputika Journal*, *9*(2), 105–113. https://doi.org/10.15294/edukomputika.v9i2.61854

Nosouhian, S., Nosouhian, F., & Khoshouei, A. K. (2021). A Review of Recurrent Neural Network Architecture for Sequence Learning: Comparison between LSTM and GRU. *Preprints*, 1–7. https://doi.org/https://doi.org/10.20944/preprints202107.0252.v1

Oktaviani, A., & Hustinawati. (2021). Prediksi Rata-Rata Zat Berbahaya di DKI Jakarta Berdasarkan Indeks Standar Pencemar Udara Menggunakan Metode Long Short-Term Memory. *Jurnal Ilmiah Informatika Komputer*, *26*(1), 41–55. https://doi.org/10.35760/ik.2021.v26i1.3702

Ouali, Y., Hudelot, C., & Tami, M. (2020). *An Overview of Deep Semi-Supervised Learning*. http://arxiv.org/abs/2006.05278

Rahma, I. A., & Suadaa, L. H. (2023). Penerapan Text Augmentation untuk Mengatasi Data yang Tidak Seimbang pada Klasifikasi Teks Berbahasa Indonesia. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *10*(6), 1329–1340. https://doi.org/10.25126/jtiik.1067325

Rolangon, A., Weku, A., & Sandag, G. A. (2023). Perbandingan Algoritma LSTM Untuk Analisis Sentimen Pengguna Twitter Terhadap Layanan Rumah Sakit Saat Pandemi Covid-19. *TeIKa*, *13*(01), 31–40. https://doi.org/10.36342/teika.v13i01.3063

Romadhoni, Y., & Holle, K. F. H. (2022). Analisis Sentimen Terhadap PERMENDIKBUD No.30 pada Media Sosial Twitter Menggunakan Metode Naive Bayes dan LSTM. *Jurnal Informatika: Jurnal Pengembangan IT*, *7*(2), 118–124. https://doi.org/10.30591/jpit.v7i2.3191

Seabe, P. L., Moutsinga, C. R. B., & Pindza, E. (2023). Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach. *Fractal and Fractional*, *7*(2), 203. https://doi.org/10.3390/fractalfract7020203

Setiyawan, D. A., W, S. K., Diana, A. L., W, I. A. H., Yusuf, M., & Krisnando, K. (2023). Penyuluhan Pemahaman Digital Wallet, Digital Perbankan Dan Pajak Penghasilan Bagi Pengusaha Kecil Untuk Meningkatkan Omzet Penjualan. *Jurnal Pengabdian Mandiri*, *2*(9), 1955–1962. https://bajangjournal.com/index.php/JPM/article/view/6615

Suryati, E., Styawati, S., & Aldino, A. A. (2023). Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Dan Sistem Informasi*, *4*(1), 96–106. https://doi.org/10.33365/jtsi.v4i1.2445

Wisnalmawati, W., Aribowo, A. S., & Herawati, Y. (2022). Semi-supervised Learning Models for Sentiment Analysis on Marketplace Dataset. *International Journal of Artificial Intelligence & Robotics (IJAIR)*, *4*(2), 78–85. https://doi.org/10.25139/ijair.v4i2.5267

Zhafira, D. F., Rahayudi, B., & Indriati, I. (2021). Analisis Sentimen Kebijakan Kampus Merdeka Menggunakan Naive Bayes dan Pembobotan TF-IDF Berdasarkan Komentar pada Youtube. *Jurnal Sistem Informasi, Teknologi Informasi, Dan Edukasi Sistem Informasi*, *2*(1). https://doi.org/10.25126/justsi.v2i1.24

Zhou, Z. H. (2021). Machine Learning. In *Machine Learning*. Springer Nature. https://doi.org/10.1007/978-981-15-1967-3/COVER