



Analisis Sentimen Terhadap Pengaruh Minat Belanja Berdasarkan Komentar di *Marketplace* Menggunakan Metode *Recurrent Neural Network (RNN)*

Gerry Santos Lasatira^{a,*}, Kristoko Dwi Hartomo^b, dan Irwan Sembiring^b

^a Magister Sistem Informasi, Universitas Kristen Satya Wacana, Salatiga, Indonesia

^b Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga, Indonesia

Naskah Diterima : 18 Maret 2023; Diterima Publikasi : 10 Oktober 2023

DOI : 10.21456/vol13iss2pp112-119

Abstract

Product reviews on the marketplace can provide useful information if they are properly processed. Product review analysis can be performed by merchants to obtain information that can be used to evaluate products and services. It is not enough to look at the number of stars in product review analysis activities; it is also necessary to look at the entire contents of the review comments to determine the intent of the review. This can be done manually in small quantities, but in large quantities, the system is more efficient. In order to understand the intent of the reviews, a system that can effectively analyze many reviews is required. Using the Recurrent Neural Network (RNN) method, this study aims to analyze sentiment on the influence of shopping interest based on comments in the marketplace. The RNN model is trained to recognize positive and negative sentiments using data from the marketplace. The sentiment analysis results will be used to assess the impact on user shopping interest in the marketplace. Sentiment analysis was performed in this study using the RNN method in the GRU/LSTM training model with epochs. The researcher determined the epoch to achieve high accuracy. The data used for model training and testing is separated into training and testing data before it is used. A comparison of 80% of training data and 20% of test data is used to split data. This study uses a training model with 77 epochs and a batch size of 128 to create a system that automatically calculates comment sentiment in the marketplace with a 100% accuracy value and determines positive and negative sentiments.

Keywords: RNN Method; Comment Data; Marketplace; GRU/LSTM; Wordcloud.

Abstrak

Ulasan produk di *marketplace* merupakan informasi yang berharga apabila diolah dengan baik. Penjual dapat melakukan analisis ulasan produk untuk mendapat informasi yang dapat digunakan dalam evaluasi produk dan layanan. Kegiatan analisis ulasan produk tidak cukup dengan melihat jumlah bintang, diperlukan melihat seluruh isi komentar ulasan untuk dapat mengetahui maksud dari ulasan. Apabila dalam jumlah sedikit dapat dilakukan secara manual, namun dalam jumlah banyak lebih efektif menggunakan sistem. Dibutuhkan sistem yang mampu menganalisis banyak ulasan dengan efektif agar memudahkan dalam memahami maksud ulasan. Penelitian ini bertujuan untuk menganalisis sentimen terhadap pengaruh minat belanja berdasarkan komentar di *marketplace* menggunakan metode *Recurrent Neural Network (RNN)*. Data komentar yang diambil dari *marketplace* digunakan untuk melatih model *RNN* dalam mengenali sentimen positif dan negatif. Hasil analisis sentimen akan digunakan untuk mengetahui pengaruhnya terhadap minat belanja pengguna di *marketplace*. Dalam penelitian ini, analisis sentiment dilakukan dengan menggunakan metode *RNN* dalam *GRU/LSTM training* model dengan *epoch*. *Epoch* ditentukan oleh peneliti untuk mendapatkan *accuracy* yang baik. Sebelum melakukan *training* dan *testing* model, data yang digunakan di *split* menjadi data *training* dan data *testing*. *Split* data dilakukan dengan perbandingan 80% data *training* dan 20% data *test*. Melalui *training* model menggunakan 77 *epoch* dan *batch size* 128 penelitian ini menghasilkan sebuah sistem yang dapat mengklasifikasikan sentimen komentar di *marketplace* secara otomatis dengan nilai *accuracy* 100%, serta dapat menentukan sentimen positif dan negatif.

Kata Kunci : Metode RNN; Data Komentar; Marketplace; GRU/LSTM; Wordcloud.

1. Pendahuluan

E-commerce adalah salah satu sektor ekonomi Indonesia yang paling dinamis dan pentingnya pembangunan didorong oleh akses Internet yang berkembang pesat. *E-commerce* diseluruh dunia didominasi dengan pangsa pasar yang besar. Ada

keuntungan dan kerugian yang terkait dengan penggunaan pasar, untuk penjual dan pembeli sama. Dalam sepuluh tahun terakhir, total ada 26,2 juta bisnis *e-commerce* di Indonesia, dengan tingkat pertumbuhan 17%. *Marketplace* menguasai mayoritas pelaku *e-commerce*. Ada banyak persaingan di pasar itu sendiri. *RNN* dalam kehidupan sehari-hari kita

*) Corresponding author: gerrylasatira14@gmail.com

sering menemui sejumlah data yang sifatnya berurutan, misalnya data teks berita, ramalan cuaca, sensor, dan video. *RNN* merupakan salah satu bentuk arsitektur *Artificial Neural Networks* (*ANN*) yang dirancang khusus untuk memproses data yang bersambung atau berurutan berurutan (*sequential data*).

RNN biasanya digunakan untuk menyelesaikan tugas yang terkait dengan data *time series*, misalnya data ramalan cuaca. Proses pengklasifikasian hanya terfokus pada satu kalimat ilmiah saja dan tidak dipengaruhi oleh kalimat sebelum dan sesudah. Dalam penelitian ini, dilakukan perbandingan antara empat metode optimasi, yaitu *Adam*, *SGD*, *Adadelta*, dan *Adamax*, untuk menentukan tingkat pembelajaran terbaik yang sesuai dengan klasifikasi kalimat. Hasil penelitian menunjukkan bahwa pengoptimalan *SGD* menghasilkan tingkat pembelajaran terbaik dengan akurasi sebesar 77,48% dan *loss* sebesar 0,71%. Metode *SGD* memiliki keunggulan karena tidak membutuhkan banyak memori *Gradient Descent* sehingga proses konvergensi menjadi lebih cepat (Putra et. al., 2017). *RNN* adalah jaringan saraf berulang atau jaringan saraf tiruan yang pemrosesannya dipanggil secara berulang-ulang untuk memproses masukan yang biasa adalah data sekuensial. Setelah mengambil suara ketukan, pemrosesan akan memasuki tahap ekstraksi fitur, *training* data, dan prediksi data.

Hasil dari proses ekstraksi fitur berupa nilai statistika, *Fast Fourier Transform* (*FFT*) dan *Mel-Frequency Cepstral Coefficients* (*MFCC*) yang kemudian akan di *training* dengan metode *RNN* dan menghasilkan *file model.h5*. Untuk hasil prediksi *precision* diatas 75%, *recall* diatas 81% dan *accuracy* diatas 87% (Tarkus et. al., 2020).

Setelah menerima barang yang dibelinya, konsumen yang melakukan pembelian melalui *marketplace* dapat menulis *review*. Beri rating dengan bintang dan tulis komentar untuk ulasan tersebut, yang meliputi tanggapan, pujian, kritik, dan masukan atas produk yang telah dibeli, membuat ulasan pembelian produk (Muktafin et.al., 2020).

E-Payment menjadi sarana utama untuk melakukan transaksi pembayaran digital. Sebagai hasilnya, peranan *e-Payment* sangat penting dalam proses jual beli di *marketplace* terutama dalam proses pembayaran. Walau begitu, banyak pengguna merasa cemas ketika harus melakukan transaksi pembayaran menggunakan *e-Payment*, terutama ketika menyangkut keamanan data dan privasi pelanggan. Salah satu keprihatinan utama adalah bagaimana menjaga keamanan data pelanggan agar tidak terancam (Nurul et. al, 2019).

Saat ini, internet memungkinkan pengguna untuk memberikan ulasan secara online di berbagai *platform*, termasuk di aplikasi *e-commerce* Shopee yang tersedia di *Google play Store*. di sana kelas sentimen positif dan negatif yang diberikan oleh

pengguna dalam ulasan *online* mencerminkan pandangan mereka terhadap berbagai jenis layanan dan produk. Selain itu, pelanggan potensial yang membaca ulasan *online* dapat sangat dipengaruhi oleh sentimen ulasan yang tertera pada kolom ulasan, baik itu bersentimen positif atau negatif. Oleh karena itu, ulasan yang ditinggalkan oleh pengguna dengan sentimen tertentu sangat mempengaruhi keputusan dalam memilih produk atau layanan yang akan digunakan (*Khafidatul and Indra, 2020*)

Analisis tentang perilaku konsumen dalam berbelanja memiliki dampak penting terhadap hobi belanja konsumen lainnya (Hartomo dan Nataliani, 2021).

Sistem peringkasan teks dengan metode abstraktif, memerlukan pendekatan *RNN* yang melakukan perhitungan bobot secara sistematis berulang kali untuk menghasilkan sistem ringkasan teks. Karena bobot yang dihitung akan lebih tepat mendekati persamaan tiap kata, *RNN* merupakan salah satu komponen dari *Deep Learning*, dapat menghasilkan nilai akurasi yang lebih unggul dari pada jaringan syaraf tiruan. *RNN* jenis *Long Short Term Memory* (*LSTM*) digunakan untuk menutupi kekurangan *RNN* yang tidak dapat menyimpan memori untuk menyortir dan menambahkan mekanisme ketertarikan sehingga setiap kata dapat lebih fokus pada konteks. Metode *stemming*, dan *non-stemming* digunakan untuk membagi pengujian berdasarkan data. Nilai ulasan tipikal berita *non-stemming* adalah 41%, akurasi 81%, dan *F-Measure* 54,27%. Sedangkan tipikal *review* berita yang menggunakan prosedur *stemming* adalah 44%, akurasi 88%, dan *F-Measure* 58,20% (Ivanedra dan Mustikasari, 2019).

Kondisi terbaik penggunaan fitur data siang dengan panjang *sequence* 20, *hidden size* 8, 3 *layer LSTM*, dan data latih 70% menghasilkan rata-rata *RMSE* sebesar 46,72, sedangkan kondisi terbaik penggunaan fitur data malam dengan panjang *sequence* 30, *hidden size* 8, 1 *layer LSTM*, dan data *training* 80% menghasilkan rata-rata *RMSE* 51,05. Ketika *RNN* dan *LSTM* dibandingkan, ditemukan bahwa *LSTM* bekerja lebih baik saat menggunakan deret waktu yang lebih lama (Selle et. al., 2022).

LSTM merupakan strategi yang dapat digunakan untuk fokus pada data deret waktu. Informasi yang digunakan dalam penelitian ini berasal dari 25 *unit* transformator yang mengumpulkan informasi dari sensor arus, tegangan, dan suhu. Untuk mengevaluasi kinerja *LSTM*, analisis kinerja menggunakan *Root Mean Squared Error* (*RMSE*) dan *Squared Correlation* (*SC*) dilakukan. Selain *LSTM*, penelitian ini juga menggunakan algoritma *Multilayer Perceptron*, *Linear Regression*, dan *Gradient Boosting Regressor* sebagai pembanding. Hasil dari uji coba menunjukkan bahwa *LSTM* memiliki kinerja yang sangat baik setelah melakukan pengaturan data, memilih tanggapan menggunakan perhitungan *KBest* dan melakukan eksplorasi berbagai variasi batas.

Dalam hal ini, strategi pembelajaran mendalam dengan *LSTM* lebih disarankan daripada ketiga perhitungan lainnya, dengan nilai *RMSE* = 0,0004 dan *Squared Correlation* = 0,9690 (Ningrum et al., 2021).

Fluktuasi daya *photovoltaic (PV)* yang disebabkan oleh perubahan cuaca dapat menyebabkan jangka pendek ketidaksesuaian permintaan dan pasokan listrik. Oleh karena itu, untuk mengoperasikan jaringan listrik secara efisien dan andal, prakiraan daya *PV* jangka pendek diperlukan terhadap fluktuasi ini. Dalam makalah ini kami mengusulkan perkiraan jangka pendek daya *PV* berbasis *RNN* yang dalam. Untuk mencerminkan dampak perubahan cuaca, model yang diusulkan menggunakan *dataset IoT*, Yang dikumpulkan secara langsung atau *real-time*. Kami menyelidiki berbagai parameter dari model perkiraan berbasis *RNN* yang diusulkan dan kombinasi parameter cuaca untuk menemukan model prediksi yang akurat. Hasil percobaan menunjukkan bahwa akurasi perkiraan pembangkit listrik *PV* 5 dan 15 menit ke depan, menggunakan 3 lapisan *RNN* dengan 12 langkah waktu, masing-masing adalah 98,0% dan 96,6% berdasarkan *RMSE* yang dinormalisasi. Pada percobaan sebelum melakukan perkiraan daya listrik *PV* selama 1 dan 3 jam, skor *R2* berturut-turut adalah 0,988 dan 0,949, dan akurasi masing-masing adalah 94,8% dan 92,9%. Selain itu, skor *R2* juga adalah 0,963 dan 0,927. Hasil percobaan menunjukkan bahwa algoritma perkiraan jangka pendek berbasis *RNN* yang diusulkan mencapai akurasi prediksi yang lebih tinggi (Ahn dan Park, 2021).

Secara keseluruhan, penggunaan jaringan syaraf tiruan tampak menjanjikan dalam pendeteksian berita palsu. Selain *CNN* dan *RNN*, arsitektur jaringan syaraf tiruan yang lebih kompleks akan dipertimbangkan sebagai bagian dari analisis kami di masa mendatang. Jika digunakan bersama dengan teknik rekayasa fitur yang dibuat khusus untuk tugas tertentu, model-model tradisional dapat memberikan manfaat yang signifikan (Nasir et al., 2021).

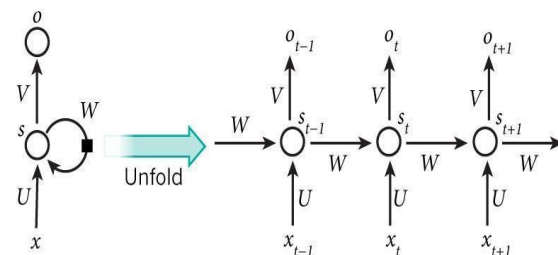
Berdasarkan penelitian yang sudah dilakukan dan uraian yang sudah dijelaskan diatas maka dilakukan penelitian dengan menggunakan salah satu metode *RNN* yang dimodifikasi yaitu *LSTM* yang digabungkan dengan *GRU*, dua metode ini digabung untuk menghasilkan analisis peningkatan penjualan produk. Tujuan dari penelitian ini adalah Analisis sentimen komentar di *marketplace*.

2. Kerangka Teori

2.1. Recurrent Neural Network (RNN)

RNN adalah salah satu cara memproses data yang bersambung (*sequential data*) dari keluarga *Neural Network*. Cara yang dilakukan *RNN* untuk dapat secara otomatis menyimpan informasi dari masa lalu adalah dengan melakukan *looping* di dalam arsitekturnya (Lestari et al., 2021).

Aktivasi *recurrent node* terdiri dari umpan balik untuk dirinya sendiri dari satu langkah waktu ke langkah berikutnya. *RNN* termasuk dalam kategori *deep learning* karena data diproses secara otomatis dan tanpa mendefinisikan fitur (Limbong et al., 2019). Secara teori *RNN* mampu menggunakan panjang urutan atau *sequence* beragam yang telah direkam sebelumnya sebagai informasi. Gambar 1. Merupakan proses *RNN*. X_t = Setiap langkah inputnya merupakan *time step*, X_1 = *corresponding* pada kata kedua dalam kalimat yang sedang diproses, S_t = *hidden state* pada setiap *time step t* "memory", O_t = *output* setiap *time step t*.



Gambar 1. Proses Menghitung RNN

2.2. Gated Recurrent Unit (GRU)

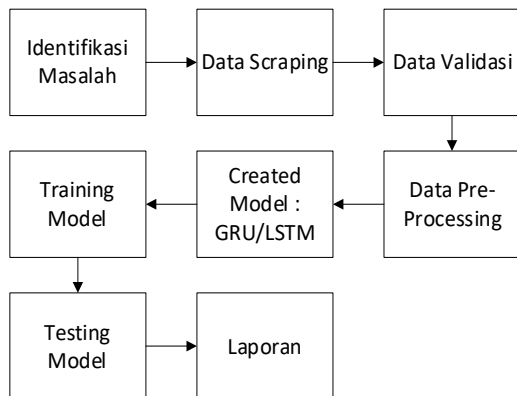
GRU adalah salah satu jenis unit dalam jaringan saraf yang berguna dalam model pemrosesan teks dan pemrosesan bahasa alami. *GRU* merupakan varian dari jenis unit lain yang disebut *Long Short-Term Memory (LSTM)*, yang berfungsi demi memproses *sequence* data yang berkelanjutan.

GRU memiliki dua jenis "gates" yaitu "*reset gate*" dan "*update gate*" yang berperan untuk mengontrol jumlah informasi yang diteruskan dari satu waktu ke waktu berikutnya. *Reset gate* bermanfaat untuk mengatur seberapa banyak informasi yang diteruskan dari waktu sebelumnya, sementara *update gate* berguna untuk mengatur seberapa banyak informasi baru yang diterima dari *input* saat ini. *GRU* dianggap lebih efisien dibanding *LSTM* karena memiliki jumlah *parameter* yang lebih sedikit dan tidak memerlukan "*output gate*" yang berfungsi dalam *LSTM*. *GRU* juga dianggap lebih mudah dipahami dan diinterpretasikan dibanding *LSTM*.

3. Metode

Pendekatan penelitian yang digunakan dijelaskan dalam Gambar 2. Prosesnya dimulai dengan pembuatan *dataset* yang berfungsi sebagai data latih. Pada tahap *pre-processing*, data diolah menjadi data yang bermakna. Kata-kata diubah menjadi vektor dalam siklus vektorisasi. Memori Jangka Pendek Panjang, atau *LSTM*, adalah teknik yang digunakan untuk membangun model. Ini adalah variasi dari *RNN*. Kemampuan model untuk mempersiapkan informasi yang sekarang dalam struktur vektor. Tes rangkuman

teks diberikan setelah latihan selesai. Evaluasi akurasi model adalah langkah terakhir.



Gambar 2. Metode Penelitian

3.1. Identifikasi Masalah

Penjual dapat menggunakan ulasan pembelian produk untuk mempelajari lebih lanjut tentang cara meningkatkan produk dan layanan mereka untuk meningkatkan kepuasan pelanggan. Tujuan perusahaan adalah untuk memastikan bahwa pelanggannya puas. Oleh karena itu, masalah sentimen yang positif dan negatif untuk penjualan produk sangat dibutuhkan, maka penelitian ini dimaksudkan agar dapat mengklasifikasikan sentimen yang positif agar dapat menguntungkan untuk membantu pelayanan *marketplace* ataupun toko yang menjual produk dalam proses jual beli.

3.2. Data Scraping

Tahapan awal Proses data *scraping* dilakukan dengan mengambil data tweet menggunakan pencarian dengan *keyword* “#ransel, #murah, #marketplace, #jual, #beli”. Data didapatkan menggunakan metode *scraping* dengan total 500 komentar di *marketplace*, adapun *range* data yaitu Januari 2022 sampai Agustus 2022. Kemudian data dipisah menjadi dua data besar sesuai dengan pendekatan pembelajaran mesin, menjadi 400 data *train* dan 100 data *test* komentar. Sampel data dapat dilihat pada Gambar 3.

No	Judul	Harga	Komentar	Bintang
1	MORYMONY TAS MISCAT LAMA - Tas Ransel Backpack Fashion Tas Sekolah	17.000	Lumayan dan bagus lah buat harga segitu sesuai harga dan barang nya.. semoga avett..	4
2	MORYMONY KEISA - Tas Ransel Backpack Fashion Wanita	15.000	Paketannya udah sampe, cepet banget pengirimannya Bagus banget dong, bahannya tebal, harganya pun murah. Thankyou seller	4
3	MORYMONY KEISA - Tas Ransel Backpack Fashion Wanita	15.000	Alhamdulillah, sesuai pesanan datang dengan lengkap dan dikirim dengan cepat lalu dtg diwaktu yg tepat	4
4	MORYMONY KEISA - Tas Ransel Backpack Fashion Wanita	15.000	Barang yg dipesan sampai dengan aman dan sesuai pesanan.	4
5	MORYMONY KEISA - Tas Ransel Backpack Fashion Wanita	15.000	Bagus banget sesuai dengan harganya ga ngecewain deh keren	4
6	MORYMONY MONICA - Tas Ransel dan Selempang 2in1	11.999	Alhamdulillah, sesuai pesanan datang dengan lengkap dan dikirim dengan cepat lalu dtg diwaktu yg tepat	4
7	MORYMONY MONICA - Tas Ransel dan Selempang 2in2	11.999	Bagus banget sesuai dengan harganya ga ngecewain deh keren	4
8	MORYMONY PAUS - Tas Ransel Backpack Korea Fashion Lucu	16.050	Semoga makin bagus ya barang untuk dijual kedepannya. Makin berkualitas walaupun murah	3
9	MORYMONY PAUS - Tas Ransel Backpack Korea Fashion Lucu	16.050	Semoga makin bagus ya barang untuk dijual kedepannya. Makin berkualitas walaupun murah	3

Gambar 3. Sampel Data

3.3. Data Validasi

Sebelum data *pre-processing*, peneliti melakukan validasi data untuk menentukan mana sentimen positif dan negatif. Data ini divalidasi oleh ahli bahasa.

3.4. Pre-processing data

Data diterapkan pada *training* data untuk membuat algoritma pembelajaran pada data tersebut. Proses ini diterapkan agar pemrosesan data lebih mudah dengan cara mendapatkan teks yang berstruktur dari *scraping* data teks yang tidak terstruktur tadi. data komentar yang menggunakan kata bahasa singkatan atau bahasa daerah akan diartikan ke Bahasa Indonesia agar data yang dihasilkan dapat disinkronisasi.

Teks *pre-processing* dilakukan memakai beberapa tahapan pada penelitian ini, dapat dilihat pada Gambar 4 dan 5. yaitu : a) *Cleansing* : adalah untuk mengurangi *noise* dengan menghapus karakter yang tidak termasuk alfabet. Karakter yang dihilangkan berupa tanda-tanda seperti “@” dan “#” masing-masing menandai *user* dan *hashtag*, url *website* dan *emoticon*. b) *Case Folding* : adalah proses menjadikan semua kalimat atau kata menjadi huruf kecil (*lower case*) pada kalimat maupun kata. c) *Tokenizing* : adalah proses menjadikan token atau ketentuan dengan memisahkan kata dari kalimat penyusun nya.

```

#Preprocessing Data

data['Komentar'].apply(lambda x: x.lower())
data['Komentar'] = data['Komentar'].apply(lambda x: re.sub("[^a-zA-Z\s]", "", x))
data['Komentar'].head()
    
```

Gambar 4. Cleansing dan Case Folding

```

tokenizer = Tokenizer(num_words=5000, split=" ")
tokenizer.fit_on_texts(data['Komentar'].values)

X = tokenizer.texts_to_sequences(data['Komentar'].values)
X = pad_sequences(X)
X[:5]
    
```

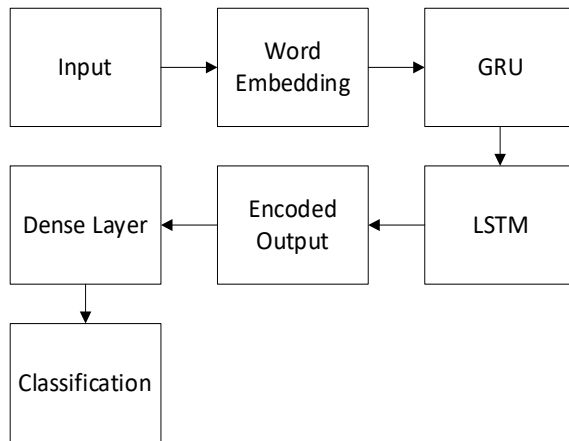
Gambar 5. Tokenizing

3.5. Creating Model GRU/LSTM

Model dibuat dengan menggunakan metode penelitian yang menggunakan jenis modul pemrosesan untuk *RNN* yaitu *GRU-LSTM*. *GRU* dan *LSTM* adalah jenis modul pemrosesan yang digunakan untuk memproses data *sequence* yang berkelanjutan dalam *RNN*. Sementara itu, *GRU* merupakan varian dari jenis unit pemrosesan lain yang disebut *LSTM*. Selain itu, data yang digunakan dalam penelitian ini dibagi menjadi dua bagian yaitu *train* data dan *test* data.

4. Hasil dan Pembahasan

Dalam penelitian ini dibangun sebuah model dengan menggabungkan kedua metode *GRU* dan *LSTM*. Proses model penelitian yang dibangun ditampilkan pada Gambar 6.



Gambar 6. Model Penelitian

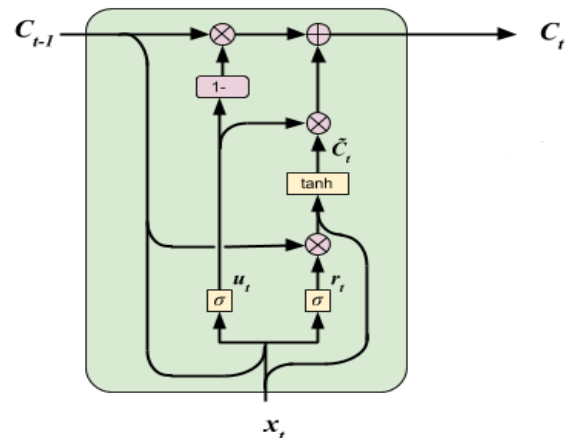
a. *Input* : Proses ini dilakukan *input dataset* dan *pre-processing* data. Proses yang membantu mengatur *dataset* dengan melakukan operasi dasar pada *dataset* sebelum meneruskannya ke model seperti penghapusan spasi dan kata yang tidak berarti, mengubah berbagai bentuk kata menjadi kata-kata akarnya, dan penghapusan kata-kata duplikat, *pre-processing* data mengubah *dataset* mentah menjadi *dataset* yang berguna dan terorganisir untuk digunakan lebih lanjut.

Word Embedding : *Dataset* yang diproses sebelumnya memberikan urutan kata yang unik dan bermakna dan setiap kata memiliki id unik. *Word Embedding* ini menginisialisasi kata-kata untuk menetapkan bobot acak dan mempelajari penyematan untuk menyematkan semua kata dalam *set* data latih. Lapisan ini digunakan dengan berbagai cara dan sebagian besar digunakan untuk mempelajari penyematan kata-kata yang dapat disimpan untuk digunakan dalam model lain.

Chung et al. (2014). Memperkenalkan *GRU* yang bertujuan untuk menciptakan recurrent unit yang dapat secara adaptif menangkap ketergantungan pada skala waktu yang berbeda. Analoginya adalah manusia tidak selalu menggunakan semua informasi masa lalu untuk membuat keputusan sekarang. Contohnya, ketika memutuskan untuk membeli makanan, informasi tentang jadwal ujian tengah semester dari masa lalu tidaklah relevan. Dengan kata lain, *GRU* dirancang agar mampu menyesuaikan informasi yang diperlukan pada saat yang tepat dan tidak memperhitungkan informasi yang tidak relevan.

Diagram arsitektur *GRU* dapat dilihat pada Gambar 7. Arsitektur ini menunjukkan kandidat konteks \tilde{c}_t melalui penggunaan fungsi *tanh* pada kombinasi antara konteks lama c_{t-1} dan *input* x_t .

Selanjutnya, gerbang *sigmoid* yang disebut sebagai gerbang *update* ut digunakan untuk menentukan seberapa besar kontribusi konteks baru c_t dalam kombinasi tersebut, dibandingkan dengan kontribusi dari konteks lama c_{t-1} . Untuk melakukan perhitungan *GRU* secara keseluruhan, ditambahkan satu gerbang lagi yang disebut sebagai *reset gate* atau *relevance gate* yang menggunakan r_t untuk menghitung nilai kandidat konteks \tilde{c}_t . dengan adanya penambahan gerbang tersebut, diagram perhitungan *GRU* menjadi lebih kompleks.



Gambar 7. Diagram Arsitektur *GRU*

Reset gate (r_t) dapat dihitung menggunakan Persamaan (1).

$$r_t = \sigma(w_r \cdot [c_{t-1}, x_t] + b_r) \quad (1)$$

dengan σ : Fungsi *Sigmoid*, w_r : Parameter *weight*, c_t : Nilai *cell state* pada *timestep* sebelumnya, x_t : Data *Input*, dan b_r : Nilai bias pada *reset gate*.

Hasil *Reset gate* (r_t) tersebut kemudian digunakan sebagai input dari *cell state* (\tilde{c}_t) yang dirumuskan pada Persamaan (2).

$$\tilde{c}_t = \tanh \tanh (w_c \cdot [r_t * c_{t-1}, x_t] + b_c) \quad (2)$$

dengan, w_c : vektor bobot untuk menghitung *current memory content*, r_t : Nilai *reset gate* pada *timestep* saat ini. c_t : Nilai *cell state* pada *timestep* sebelumnya, x_t : *Input* dalam tiap waktu dan b_c : Nilai bias.

Sedangkan persamaan *Reset Gate* (u_t) dinyatakan pada Persamaan (3).

$$u_t = \sigma(w_u \cdot [c_{t-1}, x_t] + b_u) \quad (3)$$

dengan, σ : Fungsi *Sigmoid*, w_u : Bobot untuk *layer input gate*, c_t : *Input* dalam tiap waktu, x_t : Nilai *input* pada *timestep* t, b_u : Nilai bias. Hasil perhitungan *cell state* (\tilde{c}_t) dan fungsi *sigmoid* (u_t) digunakan sebagai

input konteks pada *timestep* t , (c_t) yang dinyatakan pada Persamaan (4).

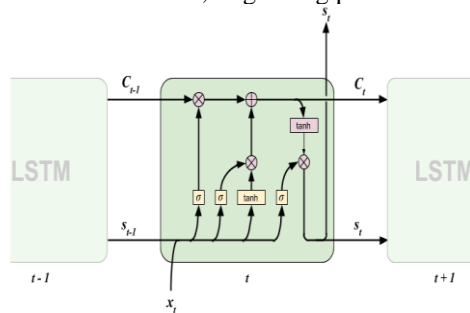
$$c_t = u_t * \tilde{c}_t + (1 - u_t) * c_{t-1} \quad (4)$$

dengan, u_t : Reset Gate, \tilde{c}_t : Nilai baru yang ditambahkan ke *cell state*, c_{t-1} : Nilai *input* pada *timestep* $t-1$.

b. Metode Long Short Term Memory (LSTM)

Setelah melakukan *max-pooling* diteruskan ke *LSTM* layer, *LSTM* menggunakan tiga jenis gerbang dan sel untuk menangani aliran informasi di jaringan. *Dropout*, karena ini mencegah model kami dari *overfitting*. Ini menjatuhkan informasi yang tidak relevan dari jaringan yang tidak berkontribusi dalam pemrosesan lebih lanjut untuk meningkatkan kinerja model kami. *Dense Layer*, lapisan padat dalam model yang diusulkan. Ini menghubungkan setiap masukan dengan setiap keluaran menggunakan bobot. *Sigmoid*, fungsi yang banyak digunakan di lapisan akhir jaringan saraf. Dibutuhkan rata-rata dari hasil acak menjadi bentuk nol (0) sampai (1). *Classification* : tahap ini merupakan akhir dari model yang dibuat, dimana sentimen positif dan negatif diklasifikasikan.

Dalam modul *LSTM*, ide utamanya adalah adanya jalur yang menghubungkan konteks lama (C_{t-1}) dengan konteks baru (C_t) di bagian atas modul *LSTM*, seperti yang terlihat pada gambar 8 di bawah ini. Konteks C_t juga disebut sebagai *cell state* atau *memory cell* dalam beberapa artikel. Dengan adanya jalur ini, nilai dari konteks lama dapat dengan mudah diwariskan ke konteks baru dengan sedikit atau tanpa modifikasi sama sekali, tergantung pada kebutuhan.



Gambar 8. Diagram Arsitektur LSTM

Arsitektur LSTM diawali dari perhitungan Forget gate (f_t) yang ditunjukkan pada Persamaan (5).

$$f_t = \sigma (w_f \cdot [s_{t-1}, x_t] + b_f) \quad (5)$$

dengan, σ : Fungsi *Sigmoid*, w_f : Nilai bobot *forget gate*, s_t : Nilai *hidden state*, x_t : *Input* dalam tiap waktu, dan b_f : Nilai bias.

Kemudian dilakukan perhitungan input gate (i_t) dengan Persamaan (6).

$$i_t = \sigma (w_i \cdot [s_{t-1}, x_t] + b_i) \quad (6)$$

dengan, σ : Fungsi *Sigmoid*, w_i : Nilai *input* pada waktu ke t , s_t : Nilai *hidden state*, x_t : *Input* dalam tiap waktu, dan b_i : Nilai bias

Nilai baru yang ditambahkan ke *cell state* (\tilde{c}_t)dirumuskan pada Persamaan (7).

$$\tilde{c}_t = \tanh (w_c \cdot [s_{t-1}, x_t] + b_c) \quad (7)$$

dengan, w_c : Nilai *input* pada *cell* ke c , s_t : Nilai *hidden state*, x_t : *Input* dalam tiap waktu, dan b_c : Nilai bias

Sedangkan nilai *memory cell state* (c_t) dinyatakan pada Persamaan 8.

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (8)$$

dengan, f_t : *Forget gate*, c_{t-1} : Nilai *memory cell state* pada *cell* sebelumnya, i_t : *Input gate*, dan \tilde{c}_t : Nilai baru yang ditambahkan ke *cell state*

Nilai dari *output gate* (o_t) diperoleh dari Persamaan 9.

$$o_t = \sigma \cdot (w_o [s_{t-1}, x_t] + b_o) \quad (9)$$

dengan, σ : Fungsi *Sigmoid*, s_t : Nilai *hidden state* dan x_t : *Input* dalam tiap waktu

Serta nilai *hidden state* (s_t) diperoleh dari Persamaan 10.

$$s_t = o_t * (\tilde{c}_t) \quad (10)$$

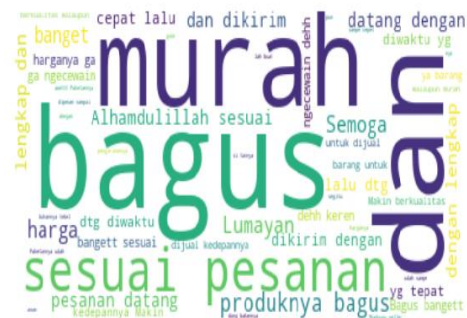
dengan, o_t : Nilai *output gate* pada *timestep* ke- t dan \tilde{c}_t : Nilai baru yang ditambahkan ke *cell state*

Dalam *GRU/LSTM training* model dilakukan dengan *epoch*. *Epoch* ditentukan oleh peneliti untuk mendapatkan *accuracy* yang baik. Sebelum melakukan *training* dan *testing* model, data yang digunakan harus di-*split* menjadi data *training* dan data *testing*. *Split* data dilakukan dengan perbandingan 80% data *training* dan 20 % data *test*. Model yang telah dilatih menggunakan 200 *epoch*, namun mencapai maksimal pada *epoch* ke-76. Oleh karena itu, dilakukan *early stopping* saat mencapai titik maksimal untuk mencegah *overfitting* pada model. *Epoch* ke-77 dihentikan karena sudah mencapai titik maksimal. Hasil *training* model menggunakan 77 *epoch* dan *batch size* 128 menghasilkan *loss*: 3,6440, *accuracy*: 1,0000, *validation_loss*: 9,3063 dan *validation_accuracy*: 1,0000, dapat kita lihat bahwa *loss* mengalami penurunan sedangkan *accuracy* mengalami kenaikan. *loss* yang merupakan *training loss* adalah perhitungan fungsi *loss* dari *training dataset* dan prediksi dari model yang dibuat. *Training accuracy* adalah perhitungan *accuracy* dari *training data set* dan prediksi modelnya. *Validation loss*

merupakan nilai penghitungan *loss function* dari *validation dataset* dan prediksi dari model dengan *input* data dari *validation data set* tersebut. *Validation accuracy* merupakan nilai penghitungan *accuracy* dari *validation dataset* dan prediksi dari model dengan *input* data dari *validation dataset*. Sedangkan *validation* sendiri adalah data atau *dataset* yang belum pernah di lihat atau di *training* dari model yang dibangun, dapat dilihat pada gambar 9. Setelah mendapatkan hasil *accuracy* yang sangat bagus, dilanjutkan dengan pengujian untuk mendapatkan kata-kata yang menentukan sentimen positif dan negatif. Kata-kata tersebut didapatkan dengan *word cloud*, dilihat pada gambar 10 dan 11.

```
Epoch 1/200
3/3 - loss: 0.6918 - accuracy: 0.4767 - val_loss: 0.6732 - val_accuracy: 0.5690 - 8s/epoch - 3s/step
Epoch 2/200
3/3 - loss: 0.6593 - accuracy: 0.5690 - val_loss: 0.6080 - val_accuracy: 0.5690 - 53ms/epoch - 184ms/step
Epoch 3/200
3/3 - loss: 0.5858 - accuracy: 0.6279 - val_loss: 0.5187 - val_accuracy: 0.6620 - 59ms/epoch - 199ms/step
Epoch 4/200
3/3 - loss: 0.5749 - accuracy: 0.6163 - val_loss: 0.4729 - val_accuracy: 0.7326 - 60ms/epoch - 208ms/step
Epoch 5/200
3/3 - loss: 0.4764 - accuracy: 0.7326 - val_loss: 0.4668 - val_accuracy: 0.7442 - 58ms/epoch - 194ms/step
Epoch 6/200
3/3 - loss: 0.4534 - accuracy: 0.7558 - val_loss: 0.4166 - val_accuracy: 0.8372 - 58ms/epoch - 194ms/step
Epoch 7/200
3/3 - loss: 0.4131 - accuracy: 0.8256 - val_loss: 0.3762 - val_accuracy: 0.8408 - 57ms/epoch - 192ms/step
Epoch 8/200
3/3 - loss: 0.3681 - accuracy: 0.8408 - val_loss: 0.3280 - val_accuracy: 0.8953 - 56ms/epoch - 189ms/step
Epoch 9/200
3/3 - loss: 0.3190 - accuracy: 0.8665 - val_loss: 0.2734 - val_accuracy: 0.8837 - 57ms/epoch - 190ms/step
Epoch 10/200
3/3 - loss: 0.2823 - accuracy: 0.8721 - val_loss: 0.2356 - val_accuracy: 0.8837 - 61ms/epoch - 200ms/step
Epoch 11/200
3/3 - loss: 0.2278 - accuracy: 0.9070 - val_loss: 0.1894 - val_accuracy: 0.9100 - 58ms/epoch - 195ms/step
Epoch 12/200
3/3 - loss: 0.1830 - accuracy: 0.9186 - val_loss: 0.1520 - val_accuracy: 0.9535 - 59ms/epoch - 199ms/step
Epoch 13/200
...
Epoch 76/200
3/3 - loss: 2.6000e-04 - accuracy: 1.0000 - val_loss: 1.0290e-04 - val_accuracy: 1.0000 - 59ms/epoch - 199ms/step
Epoch 77/200
3/3 - loss: 3.6440e-04 - accuracy: 1.0000 - val_loss: 9.3062e-05 - val_accuracy: 1.0000 - 60ms/epoch - 201ms/step
```

Gambar 9. Hasil Model Training



Gambar 10. Word Cloud Sentimen Positif



Gambar 11. Word Cloud Sentimen Negatif

5. Kesimpulan

Penelitian ini telah dilakukan analisis sentimen terhadap pengaruh minat belanja berdasarkan komentar di *marketplace*. Analisis ini dilakukan dengan menggunakan metode *RNN* untuk mengevaluasi komentar dan menentukan tingkat sentimen yang terkait dengan minat belanja. *RNN* dalam *GRU/LSTM* digunakan untuk melakukan *training* model dengan *epoch* yang ditentukan untuk mencapai *accuracy* yang baik. Data yang digunakan dibagi menjadi data *training* dan data *testing* dengan perbandingan 80% dan 20%. Hasil *training* model menggunakan 77 *epoch* dan *batch size* 128 menghasilkan *accuracy* sebesar 100%. serta mendapatkan kata-kata yang menentukan sentimen positif dan negatif.

Daftar Pustaka

- Ahn, H. K. dan Park, N. (2021) 'Deep *RNN*-Based Photovoltaic Power Short-Term Forecast Using Power *IoT* Sensors', *Energies*, 14(2), pp. 1–17. Doi: 10.3390/En14020436.
- Hartomo, K. D. dan Nataliani, Y. (2021) 'A New Model For Learning-Based Forecasting Procedure By Combining K-Means Clustering And Time Series Forecasting Algorithms', *Peerj Computer Science*, 7, P. E534. Doi: 10.7717/Peerj-Cs.534.
- Ivanedra, K. dan Mustikasari, M. (2019) 'Implementasi Metode Recurrent Neural Network Pada Text Summarization Dengan Teknik Abstraktif', *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(4), P. 377. Doi: 10.25126/Jtiik.2019641067.
- Limbong, J.J.A., Sembiring, I. dan Hartomo, K.D., (2019) 'Analisis Klasifikasi Sentimen Ulasan Pada *E-commerce* Shopee Berbasis Word Cloud Dengan Metode Naive Bayes Dan K-Nearest Neighbor Analysis Of Review Sentiment Classification On *E-commerce* Shopee Word Cloud Based With Naive Bayes And K-Nearest Neighbor Meth', *Jurnal Teknologi Informasi Dan Ilmu Komputer (Jtiik)*, 9(2), pp. 347–356. Doi: 10.25126/Jtiik.2s02294960.
- Khafidatul, I. And Indra, K. (2020) 'Pengaruh Ulasan Produk, Kemudahan, Kepercayaan, Dan Harga Terhadap *Marketplace* Shopee Di Mojokerto', *Jurnal Manajemen*, 6(1), Pp. 31–42. Available At: [Http://Www.Maker.Ac.Id/Index.Php/Maker](http://Www.Maker.Ac.Id/Index.Php/Maker).
- Lestari, M., Purnomo, H. D. And Sembiring, I. (2021) 'Pengaruh *E-Payment* Trust Terhadap Minat Transaksi Pada *E-marketplace* Menggunakan Framework Technology Acceptance Model (Tam 3)', *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 8(5), P. 977. Doi: 10.25126/Jtiik.2021855212.
- Muktafin, E. H., Kusri, K. And Luthfi, E. T. (2020) 'Analisis Sentimen Pada Ulasan Pembelian

- Produk Di *Marketplace* Shopee Menggunakan Pendekatan Natural Language Processing', *Jurnal Eksplora Informatika*, 10(1), pp. 32–42. Doi: 10.30864/Eksplora.V10i1.390.
- Nasir, J. A., Khan, O. S., dan Varlamis, I. (2021) 'Fake News Detection: A Hybrid *CNN-RNN* Based Deep Learning Approach', *International Journal Of Information Management Data Insights*, 1(1). Doi: 10.1016/J.Jjimei.2020.100007.
- Ningrum, A.A., Syarif, I., Gunawan, A.I., Satriyanto, E. dan Muchtar, R. (2021) 'Algoritma Deep Learning-*LSTM* Untuk Memprediksi Umur Transformator', *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 8(3), P. 539. Doi: 10.25126/Jtiik.2021834587.
- Nurul, M., Soewarno, N. dan Isnalita, I. (2019) 'Pengaruh Jumlah Pengunjung, Ulasan Produk, Reputasi Toko Dan Status Gold Badge Pada Penjualan Dalam Tokopedia', *E-Jurnal Akuntansi*, 28(3), p. 1855. Doi: 10.24843/Eja.2019.V28.I03.P14.
- Putra, A. K., Nyoto, R. D. dan Sasty, P. H. (2017) 'Rancang Bangun Aplikasi *Marketplace* Penyedia Jasa Les Private Di Kota Pontianak Berbasis Web', *Jurnal Sistem Dan Teknologi Informasi (Justin)*, 5(1), pp. 22–26. Available At: <Http://Jurnal.Untan.Ac.Id/Index.Php/Justin/Article/Viewfile/17991/15281>.
- Selle, N., Yudistira, N. And Dewi, C. (2022) 'Perbandingan Prediksi Penggunaan Listrik Dengan Menggunakan Metode *Long Short Term Memory (LSTM)* Dan *Recurrent Neural Network(RNN)*', *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 9(1), P. 155. Doi: 10.25126/Jtiik.2022915585.
- Tarkus, D., Sompie, S. R. U. A. dan Jacobus, A. (2020) 'Implementasi Metode *Recurrent Neural Network* Pada Pengklasifikasian Kualitas Telur Puyuh', *Jurnal Teknik Informatika*, 15(2).