

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian yang membahas mengenai analisis kepuasan pelanggan berdasarkan ulasan *online* telah banyak dilakukan dengan menerapkan metode *machine learning* maupun *deep learning*. Penelitian (Rao, dkk., 2018) mengimplementasikan *sentence representation-LSTM* dengan dua *hidden layer* memodelkan hubungan semantik antar kalimat dan klasifikasi sentimen tingkat dokumen. Tiga data set yang digunakan berasal dari YELP 2014, YELP 2015 dan IMDB. Glove 300-dimensi digunakan untuk melatih word *embedding*. Proses tokenisasi dan pembagian kalimat menggunakan Stanford CoreNLP. Dataset digunakan untuk pelatihan validasi dan pengujian dengan pembagian masing-masing data 80/10/10. Hasil klasifikasi menunjukkan bahwa model yang diusulkan memiliki kemampuan akurasi yang baik sebesar 0.639 yaitu 0.015 lebih baik dari CLSTM dan 0.026 persen lebih baik dari 2-layer LSTM.

Penelitian yang dilakukan oleh (Song, dkk, 2019) mengusulkan metode *embedding* leksikon sentimen untuk analisis sentimen tingkat aspek dalam bahasa Korea. Vektor kata dipoleh dari model word2vec dengan format input yang telah disesuaikan dengan memetakan morfem dan tag *part of speech* (POS). Penggabungan vektor aspek dan vektor kata dilakukan untuk menyatukan informasi opini target. Pada penelitian tersebut, *attention-based LSTM* digunakan untuk menggabungkan vektor aspek dengan vektor leksikon sentimen yang diusulkan. Hasil dari pendekatan yang diusulkan menunjukkan mampu mengurangi masalah pada metode *conventional context-based word embedding* dan meningkatkan kinerja klasifikasi sentimen menggunakan bahasa Korea.

RNN-LSTM dan algoritma *Doc2Vec* digunakan untuk mempelajari aspek yang mempengaruhi pilihan pengguna. Hasil penelitian tersebut menunjukkan bahwa RNN-LSTM memiliki kemampuan dalam menyimpan dan mengakses pengetahuan pilihan pengguna sehingga menghasilkan ringkasan yang akurat dari pilihan pengguna dan aspek item yang sesuai berdasarkan ulasan. Selain itu,

penerapan Doc2Vec menghasilkan representasi numerik dari umpan balik yang terdiri dari konteks semantik dan keterkaitan antara pengguna dan *item* yang merupakan salah satu dasar untuk menentukan pilihan pengguna. Pendekatan yang diterapkan mampu mengungguli metode CNN-D-att dan CNN ConvMF+. Metode ini juga mengungguli dua metode lainnya yaitu BoWLF dan LMLF yang menggabungkan model faktorisasi matriks dengan dengan LSTM-RNN dan metode kumpulan kata. Pendekatan yang diusulkan mengungguli model *Ratings Meets Reviews*(RMR) yang menerapkan model topik pada teks ulasan untuk meningkatkan prediksi model (Chambua, dkk, 2019).

Penelitian lain yang menerapkan LSTM untuk analisis sentimen berbasis aspek telah dilakukan (Al-Smadi, dkk, 2019). Penelitian tersebut mengimplementasikan dua model LSTM pada ulasan Hotel menggunakan bahasa Arab. Model LSTM pertama yaitu *character-level bidirectional LSTM* dengan *conditional random field classifier* (Bi-LSTM-CRF) untuk *opinion target expressions* (OTEs) dan LSTM berbasis aspek untuk klasifikasi polaritas sentimen dimana aspek-OTE dianggap sebagai ekspresi perhatian untuk identifikasi polaritas sentimen. Hasil dari model yang diusulkan menunjukkan bahwa terdapat peningkatan sebesar 39% untuk ekstraksi aspek-OTE dan 6% untuk klasifikasi polaritas sentimen aspek. Hasil menunjukkan bahwa pendekatan yang diusulkan mengungguli metode *baseline*. Selain itu, hasil yang diperoleh LSTM berbasis aspek untuk klasifikasi polaritas menghasilkan akurasi 82.60% lebih baik dibandingkan dengan SVM yang memiliki akurasi 81.72%.

Penelitian mengenai pemanfaatan teks ulasan untuk memahami minat pengguna telah dilakukan oleh (Shi, dkk, 2020). Dalam penelitian tersebut menerapkan algoritma *recurrent neural network* (RNN) dan *inverse document frequency* (IDF) untuk mempelajari hubungan antar kata dari jejaring sosial. Hasil penelitian tersebut menunjukkan RNN mengungguli metode lainnya seperti LDA pada hasil pengelompokan topik. Hal tersebut disebabkan RNN digunakan untuk mempelajari representasi topik dan distribusi topik dipelajari berdasarkan niat pengguna dan pengikut dari jejaring sosial.

Model yang digunakan untuk meningkatkan kemampuan dalam analisis sentimen berbasis aspek (ABSA) berdasarkan *conventional attention-based bidirectional long short-term memory network* (BiLSTM). Model untuk klasifikasi dilatih kemudian dilakukan ekstraksi fitur dan melakukan prediksi polaritas sentimen dari dataset. Selain itu, (Wu, dkk, 2019) Untuk memanfaatkan pengetahuan sentimen, informasi dalam dokumen yang memiliki label sentimen diekstraksi dan digabungkan dengan model yang diusulkan menggunakan inisialisasi parameter. Selain itu, untuk mengekstrak hubungan sentimen antar klausa diterapkan konjungsi. Hasil eksperimen yang dilakukan pada lima kumpulan dataset ABSA menunjukkan keefektifan model yang diusulkan.

Jaringan *Coattention-LSTM* diterapkan untuk mengekstrak fitur sentimen yang lebih efektif dari *coattention* dan memahami representasi kata dan target secara bersamaan. Selanjutnya, untuk meningkat hasil klasifikasi sentimen (Yang, dkk, 2019) mengadopsi mekanisme *multi-hop* pada jaringan *Coattention-MemNet*. Penelitian pada dua kumpulan data menunjukkan keefektifan metode yang diusulkan dan memberikan wawasan baru untuk pengembangan lebih lanjut pada penerapan mekanisme perhatian dan *deep neural network* untuk analisis sentimen berbasis aspek.

Recurrent neural network (RNN) yang disusun oleh *Long short-term memory* (LSTM) digunakan untuk klasifikasi opini pengguna yang diungkapkan dalam ulasan. Metode tersebut dimanfaatkan pemrosesan sekuensial dan mengatasi kekurangan metode tradisional seperti informasi dan urutan kata hilang. Untuk memverifikasi efektivitas model yang diusulkan, maka dilakukan klasifikasi sentimen tingkat kalimat pada dataset berukuran besar. Hasil eksperimen menunjukkan bahwa vektor fitur dalam pengetahuan *linguistic, statistic* dan sentimen, aturan pengubah sentimen dan penyisipan kata dapat meningkatkan akurasi klasifikasi analisis sentimen tingkat kalimat, metode yang diusulkan memiliki kinerja yang baik pada kumpulan fitur terpadu dibanding dari subset fitur dan model yang diusulkan menghasilkan peningkatan kinerja yang lebih baik dibandingkan dengan metode pendekatan lain (Abdi, dkk, 2019).

Model *Recurrent neural network* (RNN) augmented yang disebut output layer *sel-connection recurrent neural networks* (OLSRNN) diusulkan untuk mengenali target opini. Setiap input node lapisan keluaran dalam model OLSRNN tidak hanya berasal dari output lapisan tersembunyi tapi juga dari lapisan output layer pada langkah sebelumnya. Hasil eksperimen menunjukkan pendekatan yang diusulkan efektif dalam pengenalan target opini dibandingkan dengan metode dasar lainnya (Liu, dkk, 2018).

Recurrent attention LSTM digunakan untuk menemukan wilayah perhatian yang mencakup kata-kata sentimen utama. Secara bertahap mengurangi jarak perhatian dan jumlah token. Pada tahap akhir proses klasifikasi, model memanfaatkan bobot kata kunci sentimen. Selain itu penerapan loss function untuk mendapatkan kata kunci yang sesuai. Hasil percobaan menggunakan dataset dokumen IMDB, YELP dan Amazon, menunjukkan bahwa model yang diusulkan mengungguli metode lain pada tugas klasifikasi sentimen tingkat dokumen (Zhang, dkk, 2021).

Daftar publikasi hasil penelitian yang terkait dengan penelitian yang dilakukan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Daftar Penelitian Terkait

| No | Penulis | Tahun | Algoritma | Keterangan |
|----|-----------|-------|---|--|
| 1 | Rao, dkk | 2018 | <i>Sentence representation</i> LSTM dengan dua hidden layer | Menerapkan <i>sentence representation- LSTM</i> dengan dua hidden layer memodelkan hubungan semantik antar kalimat dan klasifikasi sentimen tingkat dokumen. |
| 2 | Song, dkk | 2019 | <i>Word2vec, POS tag, dan attention-based LSTM.</i> | mengusulkan metode <i>embedding</i> leksikon sentimen untuk analisis sentimen tingkat aspek dalam bahasa Korea. <i>Attention-based LSTM</i> digunakan untuk menggabungkan vektor aspek dengan vektor leksikon sentimen yang diusulkan. |

Tabel 2.1 Daftar Penelitian Terkait (lanjutan)

| No | Penulis | Tahun | Algoritma | Keterangan |
|----|---------------|-------|---|---|
| 3 | Chambua, dkk | 2019 | RNN-LSTM dan Doc2Vec. | RNN-LSTM diterapkan untuk mempelajari pengetahuan pilihan pelanggan dengan aspek item yang mempengaruhi pilihan pelanggan. Sementara itu, algoritma Doc2Vec digunakan untuk mengubah ringkasan item dan teks ulasan menjadi representasi numerik (vektor) dengan mempertahankan makna penting dari teks ulasan. |
| 4 | Al-Smadi, dkk | 2019 | LSTM untuk <i>aspect-based sentiment analysis</i> . | Metode yang diusulkan memiliki akurasi 82.60% lebih baik dibandingkan SVM yang memiliki akurasi sebesar 81.72%. |
| 5 | Shi, dkk | 2020 | <i>Recurrent Neural Network (RNN)</i> dan <i>Inverse Document Frequency (IDF)</i> | RNN dan IDF digunakan untuk mempelajari hubungan antar kata di jejaring sosial sehingga didapatkan topik yang mewakili preferensi dan minat pengguna. |
| 6 | Wu, dkk | 2019 | <i>conventional attention-based bidirectional long short-term memory network (BiLSTM)</i> | Menerapkan metode BiLSTM untuk meningkatkan kemampuan dalam analisis sentimen berbasis aspek (ABSA) |
| 7 | Yang, dkk | 2019 | <i>Coattention-LSTM</i> | Jaringan <i>Coattention-LSTM</i> diterapkan untuk mengekstrak fitur sentimen yang lebih efektif dari <i>coattention</i> dan memahami representasi kata dan target secara bersamaan. |

Tabel 2.1 Daftar Penelitian Terkait (lanjutan)

| No | Penulis | Tahun | Algoritma | Keterangan |
|----|-------------|-------|---|--|
| 8 | Abdi, dkk | 2019 | <i>Recurrent neural network</i> (RNN), <i>Long short-term memory</i> (LSTM) | Melakukan klasifikasi opini pengguna yang diungkapkan dalam ulasan. |
| 9 | Liu, dkk | 2018 | <i>Recurrent neural network</i> (RNN) augmented | <i>Recurrent neural network</i> (RNN) augmented diusulkan untuk mengenali target opini. |
| 10 | Zhang, dkk, | 2021 | <i>Recurrent attention LSTM</i> | <i>Recurrent attention LSTM</i> digunakan untuk menemukan wilayah perhatian yang mencakup kata-kata sentimen utama |

Penelitian ini mengusulkan metode LSTM dengan metode representasi vektor menggunakan CBOW yang merupakan salah satu arsitektur Word2Vec. LSTM merupakan metode yang banyak diimplementasikan pada studi kasus *Netural Language Processing* karena memiliki kinerja yang baik dalam klasifikasi teks karena mampu merepresentasikan teks atau kalimat secara berurutan, Word2vec digunakan untuk mengubah format teks menjadi vektor sehingga dapat diproses oleh LSTM. CBOW dipilih karena memiliki kemampuan representasi kata yang baik dan kecepatan pada saat proses pelatihan. Dengan menerapkan metode tersebut diperoleh model klasifikasi yang baik untuk analisis sentimen berbasis aspek. Adapun Tabel 2.2 menunjukkan penelitian yang dilakukan.

Tabel 2.2 Penelitian yang Dilakukan

| Penelitian Saat ini | Algoritma | Keterangan |
|--|------------------------|---|
| Analisis sentimen berbasis aspek untuk analisis kepuasan pengguna aplikasi VoD | LSTM, Word2vec dan SGD | Membangun model LSTM yang diimplementasikan pada analisis sentiment berbasis aspek untuk memahami kepuasan pengguna aplikasi Netflix. |

2.2 Dasar Teori

2.2.1 Aspect-Based Sentiment Analysis

Analisis sentimen merupakan salah satu bidang yang banyak diterapkan dalam bisnis karena saat ini pelanggan dapat mengungkapkan pendapat terhadap produk atau layanan melalui media sosial, *e-commerce* dan situs ulasan *online*. Hal ini dikarenakan analisis sentimen dimanfaatkan untuk memahami suara pelanggan suatu produk atau layanan. Analisis sentimen menentukan sentimen positif, negatif atau netral berdasarkan kumpulan ulasan yang diberikan pengguna. Perusahaan dapat meningkatkan layanan atau produk yang mereka, selain itu dapat pula digunakan untuk melakukan analisis terhadap produk dari perusahaan pesaing.

Aspect-based sentiment analysis (ABSA) merupakan teknik yang digunakan untuk melakukan analisis aspek produk atau layanan berdasarkan ulasan pelanggan (de Kok, dkk, 2018). Sebagai contoh, “pembayarannya ribet cuma bisa pake kartu kredit”, yang mengungkapkan sifat negatif pada aspek “metode pembayaran”. Saat ini, Ketika pelanggan ingin membeli atau menggunakan produk seringkali memperhatikan ulasan yang diberikan pelanggan lain yang lebih dahulu sebagai bahan pertimbangan sebelum membeli. ABSA diterapkan untuk melakukan ekstraksi fitur penting yang disebut aspek pada suatu ulasan pelanggan karena mengetahui positif, negatif atau netral saja tidak cukup (Alamanda, 2020). ABSA yang diterapkan pada analisis ulasan produk berguna sebagai bahan rekomendasi untuk menentukan menentukan fitur produk yang perlu diperbaiki. Ekstraksi aspek memiliki peran penting di ABSA yaitu untuk mengidentifikasi objek opini pengguna dari ulasan *online* (Rintyarna, dkk, 2019).

2.2.2 Word Embedding

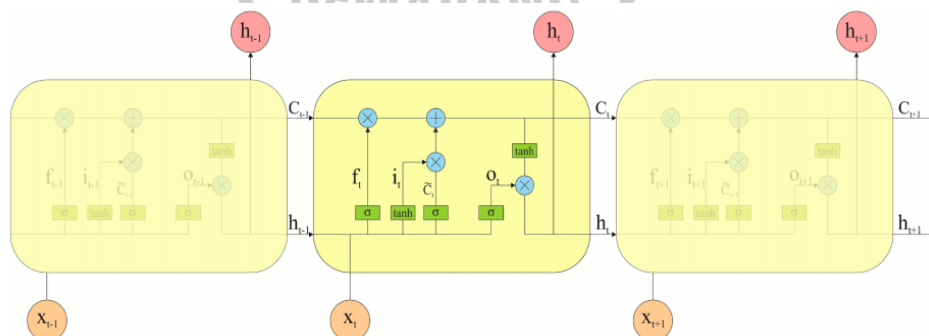
Word embedding telah banyak diterapkan dalam pemrosesan bahasa alami karena memiliki kemampuan dalam representasi vektor yang mampu menangkap kesamaan antar kata (Y. Wang, dkk, 2018).). Untuk membangun model NLP pada *deep learning* kata-kata perlu diubah terlebih dahulu menjadi vektor. Proses perubahan kata-kata menjadi vektor disebut *word embedding* atau *embedding* kata. Dimana kata-kata yang memiliki arti yang mirip dikelompokkan kedalam posisi

yang berdekatan dalam volume ruang yang sama (Pasupa dan Seneewong Na Ayutthaya, 2019).

Word2vec dikembangkan (Mikolov, dkk, 2013) merupakan salah satu metode embedding kata untuk mengolah kata-kata dari dataset yang besar menggunakan model *neural networks*. Metode ini akan menghasilkan representasi vektor dari setiap kata. *Word2vec* memiliki dua jenis model arsitektur yang dapat digunakan untuk merepresentasikan vektor kata yaitu *skip-gram* dan *continous bag-of-words* (CBOW) (Mikolov, dkk, 2013). Model arsitektur *skip-gram* memprediksi kata target berdasarkan konteks kata (*output*) di sekitar *current word* (*input*), sedangkan CBOW untuk melakukan prediksi kata (*output*) disekitar kata tersebut (*input*). *Skip-gram* dapat berfungsi lebih baik untuk jumlah teks pelatihan yang lebih sedikit, sedangkan CBOW memiliki keunggulan kualitas representasi kata yang lebih tinggi dan peningkatan kecepatan pelatihan.

2.2.3 Long Short-Term Memory

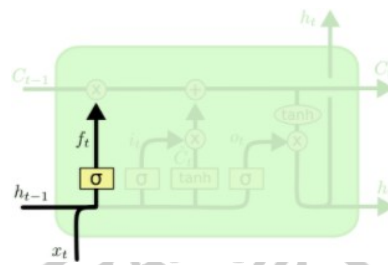
Long Short-Term Memory (LSTM) merupakan salah satu jenis dari *recurrent neural networks* (RNN) yang memiliki kelebihan untuk memprediksi dan mempelajari data sekuensial. LSTM memiliki struktur memori yang dapat mengingat informasi untuk jangka waktu yang lama dan tiga gerbang. Komponen utama lapisan LSTM disebut sebagai sel memori (Chen, dkk, 2020).



Gambar 2.1 Struktur LSTM (Chen, dkk, 2020)

Pada struktur LSTM Gambar 2.1 gerbang LSTM terdiri dari *input gate* (i_t), *forget gate* (f_t) dan *output gate* (o_t) untuk mengatur aliran informasi masuk dan keluar dari sel (Al-jabery, dkk, 2020). Pengimplementasian LSTM di beberapa aplikasi

menunjukkan hasil yang baik, khususnya pada aplikasi pengenalan ucapan, sintesis ucapan, pemodelan bahasa dan terjemahan, dan pengenalan tulisan tangan (Al-jabery, dkk, 2020). Formula yang digunakan untuk model aspek-level klasifikasi sentimen dinotasikan pada persamaan (2.1) sampai (2.6). Formula yang digunakan untuk menghitung *forget gate* dinotasikan pada formula (2.1).

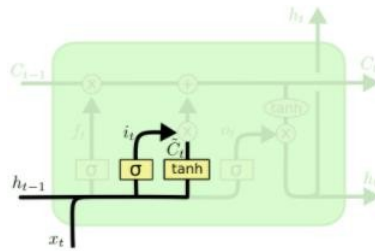


Gambar 2.2 *Forget Gate* (Olah, 2015)

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2.1)$$

Dengan f_t merupakan *forget gate*, σ fungsi sigmoid, W_f nilai bobot pada forget gate, x_t data masukkan pada waktu ke- t , U_f nilai matrix pada forget gate, h_{t-1} nilai keluaran pada tahap sebelumnya dan b_f adalah bias. Proses yang dilakukan pada Gambar 2.2 merupakan tahap pertama pada LSTM yaitu untuk menentukan informasi atau data dihapus atau disimpan oleh *cell state* (C_{t-1}) menggunakan sigmoid.

Terdapat dua data masukan yaitu data masukan pada waktu tertentu (x_t) dan data keluaran pada sel sebelumnya (h_{t-1}) dan dikalikan dengan matrix bobot (U_f) diikuti dengan penambahan bias. Keluarannya dimasukkan ke dalam sebuah fungsi aktivasi yang akan menghasilkan *output* biner (σ). Jika hasil dari *cell state* berupa nilai 0 maka informasi tersebut akan dihilangkan atau dilupakan dan jika hasil memiliki nilai 1, informasi tersebut akan disimpan untuk digunakan pada tahap berikutnya.



Gambar 2.3 Menentukan Informasi Baru pada *Cell State* (Olah, 2015)

Langkah selanjutnya, tahapan untuk menentukan informasi baru yang harus disimpan pada status sel ditunjukkan pada Gambar 2.3. Pertama, lapisan yang disebut sebagai lapisan gerbang masukkan untuk menentukan nilai mana yang diperbaharui dengan menggunakan formula 2.2 informasi diolah melalui *input gate* (i_t).

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2.2)$$

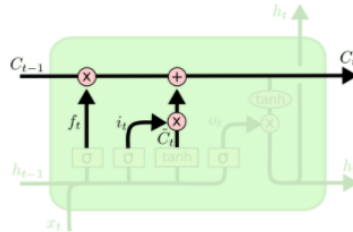
Dengan i_t *input gate*, σ fungsi sigmoid, W_i nilai bobot pada *input gate*, x_t data *input* pada waktu ke- t , h_{t-1} nilai keluaran dari langkah waktu sebelumnya dan b_i adalah bias. Proses yang dilakukan pada tahap ini akan menentukan informasi yang akan diperbaharui pada status sel menggunakan fungsi aktivasi sigmoid (σ) dan menyaring nilai yang akan disimpan dengan *forget gate* menggunakan masukan (h_{t-1}) dan (x_t).

Selanjutnya, lapisan *tanh* yang membuat nilai pada kandidat baru \tilde{C}_t (formula 2.3).

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (2.3)$$

Dengan \tilde{C}_t nilai baru, *tanh* fungsi *tanh*, W_c nilai bobot pada *cell state*, x_t nilai masukkan pada waktu ke- t , U_c nilai matrik pada *input gate*, h_{t-1} merupakan nilai keluaran dari langkah waktu sebelumnya dan b_c adalah bias. Nilai yang dihasilkan oleh lapisan *tanh* diantara -1 dan 1 yang mengandung semua kemungkinan nilai dari (h_{t-1}) dan (x_t). Setelah itu, nilai-nilai tersebut dikalikan untuk mendapatkan informasi penting.

Setelah menentukan informasi baru yang harus disimpan pada status sel, langkah selanjutnya adalah memperbarui status sel sebelumnya c_{t-1} menjadi status sel baru c_t .



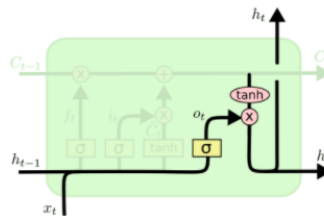
Gambar 2.4 Memperbaharui Nilai *Cell State* (Olah, 2015)

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{C}_t \quad (2.4)$$

Dengan c_t *cell state*, f_t *forget gate*, c_{t-1} status sel sebelumnya, i_t *input gate*, \tilde{C}_t nilai baru, dan \circ merupakan operasi perkalian. Pada langkah sebelumnya c_{t-1} dikalikan dengan f_t , hasil perkalian ditambahkan dengan i_t dikali \tilde{C}_t . Ini dilakukan untuk memperbaharui nilai di setiap *state* yang merupakan proses pada Gambar 2.4.

Sekolah Pascasarjana

Pada langkah terakhir, gerbang keluaran o_t melakukan tugas untuk mengekstrak informasi yang berguna dari *cell state* sebagai hasil Gambar 2.5.



Gambar 2.5 Output Gate (Olah, 2015)

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (2.5)$$

Dengan o_t *output gate*, σ fungsi sigmoid, W_o nilai bobot pada *output gate*, x_t data masukan pada waktu ke- t , U_o nilai matrik pada *output gate*, h_{t-1} nilai keluaran pada langkah waktu sebelumnya dan b_o adalah bias.

Langkah pertama, fungsi aktivasi sigmoid digunakan untuk memutuskan informasi yang akan dihasilkan. Selanjutnya, nilai keluaran *cell state* dimasukkan ke lapisan *tanh* (berfungsi sebagai nilai pengganti antara 1 dan -1) kemudian dikalikan dengan sigmoid untuk memutuskan informasi yang akan disimpan menggunakan data *input* (h_{t-1}) dan (x_t) sehingga diperoleh informasi untuk dikirimkan sebagai hasil dan data masukan pada sel selanjutnya. Untuk menentukan gerbang keluaran dapat menggunakan formula (2.5). Selanjutnya adalah menghitung hasil keluaran yang dimasukkan ke dalam fungsi *tanh* lalu dikalikan dengan nilai keluaran dari gerbang *sigmoid* menggunakan formula 2.6.

$$h_t = o_t \circ \tanh(c_t) \quad (2.6)$$

Dengan h_t adalah *hidden state*, o_t *output gate*, \tanh merupakan fungsi *tanh*, \circ operasi perkalian dan c_t adalah status sel.

Pada penelitian sebelumnya, *output* dari setiap kalimat digunakan untuk mewakili kalimat. Representasi kalimat kemudian digunakan sebagai fitur untuk klasifikasi sentimen. *Softmax* digunakan untuk melakukan klasifikasi kata sebagai positif, negatif atau netral (Rao, dkk, 2018).

Formula yang digunakan untuk menghitung *softmax* dapat dilihat pada formula (7), dimana C adalah jumlah kategori sentimen:

$$\text{softmax}_i = \frac{\exp(x_i)}{\sum_{i=0}^C \exp(x_i^o)} \quad (2.7)$$

Pada penelitian ini menerapkan *categorical cross-entropy* untuk melakukan pelatihan model sebagai *loss function* dan metode *stochastic gradient descent* (SGD). Tujuan dari pelatihan yaitu meminimalkan kerugian *categorical cross-entropy* (Zhou, dkk, 2016). L_2 regularization digabungkan ke semua parameter untuk mengurangi *over-fitting*. *Over-fitting* terjadi saat model salah membagi data training, termasuk data *noise* sehingga diperoleh biaya rendah (Rao, dkk, 2018).

L_2 regularization yang terdapat pada formula 2.8 dapat membatasi ukuran bobot dan membuat model tidak dapat menyesuaikan *noise* acak dalam data pelatihan secara acak (Liu, dkk, 2020). Pada penelitian yang dilakukan sebelumnya, L_2 regularization diterapkan sebagai solusi untuk mengatasi *overfitting*.

$$\text{loss} = -\sum_i \sum_j y_i^j \log g_j^j + \frac{\lambda \|\theta\|^2}{2} \quad (2.8)$$

Dengan y adalah distribusi target untuk setiap kalimat, z adalah distribusi kalimat yang diprediksi. Tujuan dari pelatihan adalah untuk meminimalkan kesalahan *cross-entropy* antara y dan z pada semua data pelatihan. Dimana I adalah indeks dari dokumen, j adalah indeks kelas. λ adalah L_2 regularization term. θ adalah set parameter (Rao, dkk, 2018).

2.2.4 Fungsi Sigmoid

Fungsi sigmoid diterapkan dalam pembelajaran mesin untuk *logistic regression* dan implementasi jaringan syaraf dasar. Nilai keluaran dari fungsi sigmoid adalah antara 0 dan 1. Formula yang digunakan pada fungsi sigmoid ditunjukkan pada formula (2.9)

$$f(x) = \frac{1}{(1+e^{-x})} \quad (2.9)$$

Dengan $f(x)$ merupakan sebuah fungsi dan e^{-x} merupakan nilai eksponensial.

2.2.5 Fungsi Tanh

Fungsi aktivasi *tanh* digunakan untuk memetakan nilai *input* dengan rentang nilai antara -1 hingga 1. Formula yang digunakan untuk menghitung fungsi aktivasi tanh dapat dilihat pada formula (2.10).

$$f(x) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}} \right) \quad (2.10)$$

Dengan $f(x)$ merupakan sebuah fungsi dan e^{-x} merupakan nilai eksponensial.

2.2.6 Pelatihan LSTM

Proses pelatihan pada LSTM terdiri tiga tahapan yaitu *feedforward propagation*, *backpropagation* dan memperbaharui nilai parameter. Adapun penjelasan mengenai proses pelatihan penerapan LSTM, yaitu (Song dkk., 2020):

1. Inisialisasi parameter yang akan digunakan seperti bobot (W) dan bobot berulang (U) menggunakan bilangan acak dan inisialisasi nilai bias b .
2. Melakukan proses *feedforward propagation*.
 - a. Menghitung nilai *forget gate* menggunakan formula 2.1 dan menghitung nilai fungsi aktivasi *sigmoid* menggunakan formula 2.9.
 - b. Menghitung nilai *input gate* menggunakan formula 2.2 dan menghitung nilai fungsi aktivasi *sigmoid* menggunakan formula 2.9.
 - c. Menghitung nilai kandidat *cell state* menggunakan formula 2.3 dan menghitung nilai fungsi aktivasi *tanh* menggunakan formula 2.10.
 - d. Menghitung nilai *cell state* menggunakan formula 2.4.

- e. Menghitung nilai *output gate* menggunakan formula 2.5 dan menghitung nilai fungsi aktivasi *sigmoid* menggunakan formula (2.9).
 - f. Menghitung nilai *output* akhir menggunakan formula (2.6)
 - g. Menghitung nilai Loss atau nilai error menggunakan formula (2.8).
3. Melakukan proses *backpropagation*.

- a. Menghitung nilai *derivative* dari error menggunakan formula (2.11).

$$\frac{\partial L}{\partial \hat{y}} = -(y - \hat{y}) \quad (2.11)$$

Dengan ∂ merupakan nilai derivative (turunan), y data yang dihasilkan dan \hat{y} adalah hasil prediksi.

- b. Menghitung nilai *derivative* dari *output* menggunakan formula (2.12).

$$\delta h_t = \Delta_t + \Delta h_t \quad (2.12)$$

Menghitung nilai *derivative* dari *output gate* menggunakan formula 2.13.

$$\delta o_t = \delta h_t \cdot \tanh(C_t) \cdot o_t(1 - o_t) \quad (2.13)$$

- c. Menghitung nilai *derivative* dari *cell state* menggunakan formula 2.14.

$$\delta C_t = \delta h_t \cdot o_t \cdot [1 - \tanh^2(C_t)] + \delta C_{t+1} \cdot f_{t+1} \quad (2.14)$$

- d. Menghitung nilai *derivative* dari kandidat *cell state* menggunakan formula 2.15.

$$\delta \tilde{C}_t = \delta C_t \cdot i_t \cdot [1 - \tanh^2(\tilde{C}_t)] \quad (2.15)$$

- e. Menghitung nilai *derivative* dari *input gate* menggunakan formula 2.16.

$$\delta i_t = \delta C_t \cdot \delta \tilde{C}_t \cdot i_t \cdot [1 - i_t] \quad (2.16)$$

- f. Menghitung nilai *derivative* dari *forget gate* menggunakan formula 2.17.

$$\delta f_t = \delta C_t \cdot \delta C_{t-1} \cdot f_t [1 - f_t] \quad (2.17)$$

- g. Menghitung nilai δx_t menggunakan formula 2.18.

$$\delta x_t = W_c^T \cdot \delta C_t + W_i^T \cdot \delta i_t + W_f^T \cdot \delta f_t + W_o^T \cdot \delta o_t$$

$$\delta x_t = W^T \cdot \delta gates_t \quad (2.18)$$

- h. Menghitung nilai gradien bobot (W) dari setiap gerbang LSTM menggunakan formula (2.19).

$$\delta W_{gates} = \sum_{t=0}^T (\delta gates_t \cdot x_t) \quad (2.19)$$

- i. Menghitung nilai *gradient* bobot berulang (U) dari setiap gerbang LSTM menggunakan formula (2.20)

$$\delta U_{gates} = \sum_{t=0}^{T-1} (\delta gates_{t+1} \cdot h_t) \quad (2.20)$$

- j. Menghitung nilai gradien bias (b) dari setiap gerbang LSTM menggunakan formula (2.21).

$$\delta U_{gates} = \sum_{t=0}^T (\delta gates_{t+1} \cdot h_t) \quad (2.21)$$

4. Melakukan proses pembaharuan bobot

- a. Menghitung nilai perubahan bobot menggunakan formula 2.22.

$$\delta W_{t+1} = W_t - \alpha \frac{\partial L}{\partial W_t} \quad (2.22)$$

2.2.7 Rectified Linear Unit (ReLU)

Rectified Linear Unit (ReLU) digunakan sebagai fungsi aktivasi non-linier karena dapat meningkatkan pembelajaran dan dapat mengurangi jumlah iterasi yang diperlukan untuk konvergensi dalam deep neural networks (Liu dan Guo, 2019). Adapun formula dari fungsi aktivasi ReLU dapat dilihat pada formula 2.23.

$$f(x) = (0, x) = \{x_i, \text{if } x_i \geq 0, x_i < 0\} \quad (2.23)$$

Fungsi aktivasi ReLU memiliki nilai $[0, \infty]$, jika data yang masuk ke dalam fungsi aktivasi ReLU sentimen maka nilainya akan menjadi 0. Namun, jika nilai yang masuk bernilai positif x , maka nilainya adalah x artinya jika $x \leq 0$, maka $x = 0$ dan jika $x > 0$, maka $x = x$.

2.2.8 SGD (Stochastic Gradient Descent)

Stochastic gradient descent (SGD) digunakan untuk melatih model dengan *loss function* adalah *cross-entropy* dari dari klasifikasi sentiment *supervised* (Rao , dkk, 2018). SGD telah banyak diimplementasikan untuk melatih pada berbagai studi kasus *deep neural network* (Cheridito, dkk, 2021) dan sebagai algoritma pembelajaran pada kumpulan data SGD dan variannya merupakan teknik-teknik yang telah terbukti sangat penting (Léon Bottou, 2012). Pada implementasi proses pelatihan model LSTM menggunakan algoritma SGD, bobot yang digunakan merupakan bobot yang diambil dari bobot word2vec dan tidak diambil secara acak atau *random initialized*. Hal ini dilakukan guna mempercepat proses pelatihan dan menghemat memori. Adapun *pseudo code* implementasi algoritma SGD untuk optimasi model LSTM dapat dilihat sebagai (Zhang, 2019).

Algoritma *Stochastic Gradient Descent*

{*Initial state: Training Set \mathcal{T} ; Learning Rate η ; Normal Distribution Std: σ*
{*Final state: Model Parameter θ* }

```
1: Asumsikan parameter secara  $\mathcal{T}$ entiment $\mathcal{T}$ tic atau acak
2: Asumsikan konvergensi  $tag = False$ 
3: While  $tag == False$  do
4:   Acak training set
5:   for setiap sampel data do
6:     Hitung gradient pada sampel data
7:     Update variabel
8:   end for
9:   if kondisi konvergensi berlaku then
10:     $tag = True$ 
11:  end if
12:end while
13:return model variabel
```

2.3 Evaluasi

Confusion matrix adalah tabel atau *matrix* yang menunjukkan banyak sampel model yang diklasifikasi benar dan banyak yang salah diklasifikasikan. Pada penelitian ini, metrik evaluasi merupakan faktor penting untuk menilai kinerja

klasifikasi dari algoritma. Klasifikasi *confusion matrix* yang digunakan pada penelitian ini dapat dilihat pada Tabel 2.3 (Fabian Pedregosa dkk., 2011).

Tabel 2.3 *Confussion Matrix*

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

Metrik-metrik yang dihitung pada *confussion matrix* meliputi akurasi, *recall* dan *precision* dan akurasi untuk melakukan evaluasi klasifikasi. Presisi digunakan untuk mengukur sampel kelas positif yang diklasifikasi dengan benar dan didefinisikan pada formula (2.24).

$$precision = \frac{TP}{TP+FP} \quad (2.24)$$

Dengan TP dan FP menunjukkan jumlah positif benar dan positif palsu. *Recall* digunakan untuk menghitung semua sampel positif. Adapun perhitungan *recall* dapat menggunakan formula (2.25).

$$recall = \frac{TP}{TP+FN} \quad (2.25)$$

Pada perhitungan *recall* TP dan FN menunjukkan jumlah positif benar dan 22entimen. Palsu. Selanjutnya, untuk menghitung prediksi yang benar dari jumlah total prediksi. Untuk menghitung akurasi dari prediksi yang benar dapat menggunakan formula (2.26) (Zhang dan Zhang, 2020).

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.26)$$

Dengan TP menunjukkan nilai positif benar ditambah TN yang merupakan nilai 22entimen benar di bagi total TP ditambah TN ditambah FN dan ditambah FN.