

Klasifikasi Data Kuesioner dengan Metode Recurrent Neural Network

Izza Luthfi Rais¹, Jondri²

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹izzalraise@students.telkomuniversity.ac.id, ²jondri@telkomuniversity.ac.id,

Abstrak

Metode Recurrect Neural Network (RNN) merupakan jaringan saraf buatan dengan pemrosesan yang dipanggil berulang kali. Dalam penelitian kali ini metode RNN akan digunakan untuk mengolah data kuesioner kepuasan mahasiswa Universitas Telkom berupa isian bebas. Keuntungan menggunakan metode pengolahan data seperti ini bertujuan untuk mempercepat hasil yang didapat dengan akurasi yang besar dalam waktu yang singkat. Arsitektur RNN yang digunakan untuk pengolahan data ini adalah LSTM (Long Short Term Memory). penggunaan LSTM dapat mempermudah pengolahan data berbentuk teks, karena memiliki *memory* untuk setiap kata pada dokumen. Dalam tugas akhir ini akan menganalisis akurasi yang didapatkan dengan menggunakan metode LSTM. Hasil uji yang didapatkan pada penelitian ini cukup baik melihat akurasi tertinggi pada class positive untuk ratio (50:50) sebesar 56.73%, class neutral untuk ratio (70:30) sebesar 82,49%, dan class negative untuk ratio (80:20) sebesar 79,84%.

Kata kunci: *RNN, LSTM*

Abstract

The Recurrect Neural Network (RNN) method is an artificial neural network with repeated processing. In this study the RNN method will be used to process the Telkom University student satisfaction questionnaire data in the form of free content. The advantage of using data processing methods like this aims to accelerate the results obtained with great accuracy in a short time. The RNN architecture used for processing this data is LSTM (Long Short Term Memory). the use of LSTM can facilitate the processing of data in the form of text, because it has memory for each word in the document. In this final project will analyze the accuracy obtained by using the LSTM method. The test results obtained in this study are good enough to see the highest accuracy in the positive class for the ratio (50:50) of 56.73%, the neutral class for the ratio (70:30) of 82.49%, and the negative class for the ratio (80:20) by 79.84%.

1. Pendahuluan

Latar belakang

Besarnya pengaruh serta manfaat dari sentimen analisis, menyebabkan banyak penelitian mengenai sentimen analisis berkembang dengan pesat [1]. Ada banyak perusahaan besar diluar negeri yang menggunakan sentimen analisis sebagai pengumpulan informasi tentang sentimen atau opini masyarakat tentang masalah yang terjadi disekitar. Sentimen analisis atau *opini mining* adalah suatu riset atau penelitian komputasional dari opini dan emosi yang diekspresikan secara tekstual [2]. Salah satu contoh penggunaan penggunaan sentimen analisis adalah identifikasi kecenderungan dan opini terhadap suatu objek atau hal-hal yang terjadi di sekitar kita [3]. Penggunaan sentimen analisis sangat dibutuhkan mengingat perkembangan social media yang bisa mempengaruhi opini seseorang. Dengan mengumpulkan informasi tentang opini-opini atau sentimen seseorang, maka kita bisa melakukan penelitian tentang sentimen analisis. Adapun penelitian yang sudah dilakukan terdahulu terkait dengan sentimen analisis seperti menganalisis sentimen pada twitter, yang diklasifikasikan dengan metode yang ada pada pembelajaran mesin atau biasa disebut dengan *Machine Learning* [4]. Dalam penelitian ini data yang digunakan berbentuk kuesioner. Banyak data kuesioner yang cukup sulit untuk diolah. Oleh karena itu, data pada kuesioner dipilih sebagai bahan penelitian. Pada data kuesioner diperlukan suatu alat untuk

membagi data pada dokumen tersebut. Data tersebut akan dibagi menjadi berbagai macam *Class*. Alat yang digunakan adalah aplikasi bernama *Sentistrength*. Dengan bantuan alat ini dapat dilihat nilai atau *value* dari isi kuesioner, yang nantinya akan dibagi menjadi beberapa *class*. Pembobotan atau vektorisasi juga diperlukan untuk mengubah kata menjadi bentuk vektor.

Deep Learning adalah penelitian *Machine Learning* dengan tujuan dibuatnya terinspirasi dari cara kerja otak manusia atau biasa disebut dengan *Artificial Intelligence* [5]. *Deep learning* sendiri digunakan untuk mengolah data seperti gambar, suara dan teks. Dalam hal ini penggunaan *Deep Learning* sangat membantu dalam pengolahan data seperti survei yang berbentuk kuesioner. Metode untuk melakukan analisis sentimen sangatlah beraneka ragam, diantaranya *naive bayes* atau *support vector machine (SVM)*. Dalam penelitian ini akan digunakan salah satu metode dari *deep learning* yaitu *Recurrent Neural Network (RNN)*. RNN merupakan model yang *biased*, dimana kata-kata yang dituliskan akhir kalimat menjadi lebih dominan jika dibandingkan dengan kata-kata yang dituliskan sebelumnya [6]. Dengan menggunakan metode yang sudah pernah digunakan ini, maka diharapkan dapat membantu analisis sentimen dan lebih baik dari penelitian sebelumnya.

Perumusan Masalah

Berdasarkan latar belakang telah dipaparkan maka didapatkan rumusan masalah pada tugas akhir ini, yaitu Apakah RNN dapat mengklasifikasikan data kuesioner dan bagaimana cara mengimplementasikan RNN untuk mengklasifikasikan data kuesioner.

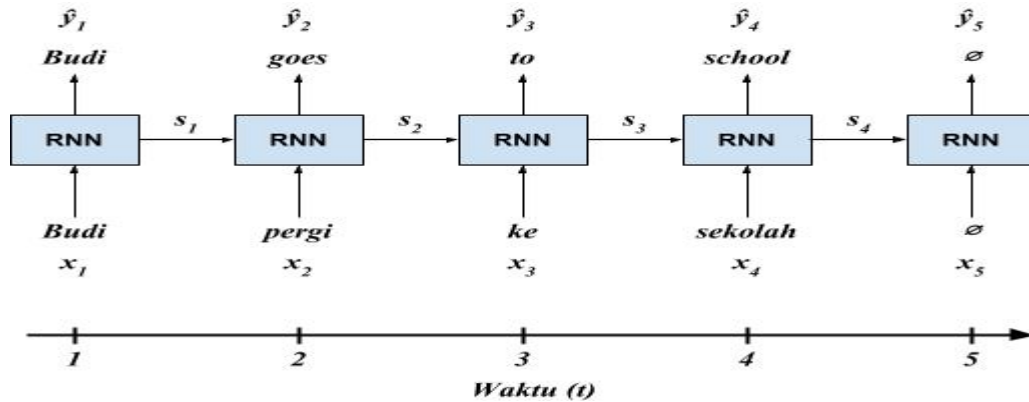
2. Studi Terkait

Sentimen Analisis

Hal yang perlu diperhatikan dalam sentimen analisis adalah mengelompokkan polaritas dari teks yang ada pada dokumen atau bahkan kalimat [1]. Pemilihan kata pada kuesioner sangatlah rumit, mengingat banyak kata yang diekspresikan oleh penulis beranekaragam. Oleh karena itu perlu diadakan analisis sentimen untuk pemilihan kata yang ada pada kuesioner. Sentimen analisis dilakukan agar opini tentang suatu objek dapat disimpulkan dengan cepat dan tepat. Analisis sentimen sendiri merupakan cara untuk menganalisa sebuah teks seperti menentukan apakah teks tersebut merupakan teks yang positif, negatif atau bahkan netral [7].

Recurrent Neural Network

Jaringan saraf tiruan berulang atau *recurrent neural network (RNN)* adalah jenis arsitektur jaringan saraf tiruan yang pemrosesannya dipanggil berulang kali untuk memproses masukan yang biasanya adalah data sekuensial. RNN merupakan bagian dari *deep learning* dimana data diproses melalui banyak lapis [7]. Data sekuensial mempunyai karakteristik dimana sampel biasanya diproses dengan suatu urutan serta suatu sampel mempunyai hubungan erat satu dengan yang lainnya. RNN memproses *input* secara sekuensial, *sampel per sampel*. Dalam setiap pemrosesan, output yang dihasilkan bisa berupa *state* internal yang merupakan hasil dari pemrosesan sampel sebelumnya atau bahkan setelahnya [7]. Berikut contoh arsitektur dari RNN.



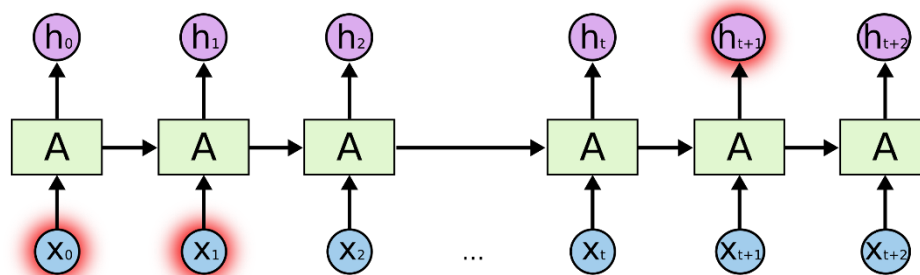
Gambar 1. Arsitektur RNN[10]

Gambar diatas menjelaskan bahwa RNN memberikan output kata demi kata yang disimbolkan dengan $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_5$, atau secara umum \hat{y}_t . Sedangkan untuk pemrosesan input oleh RNN disimbolkan dengan x_1, x_2, \dots, x_5 , atau secara umum x_t . Jaringan syaraf berulang dari RNN sama dengan precpetron multi layer namun mempunyai satu layer tersembunyi [4]. Jaringan syaraf RNN mampu mengolah suatu urutan panjang dengan menerapkan fungsi transisi secara rekursif ke vektor keadaan tersembunyi h_t dari urutan input [7]. Aktivasi h_t keadaan tersembunyi pada waktu t dihitung sebagai fungsi f dari simbol masukan x_t dan keadaan tersembunyi sebelumnya h_{t-1}

$$h_t = f(x) = \begin{cases} 0 & t = 0 \\ f(h_{t-1}, x_t) & t > 0 \end{cases} \quad (1)$$

Long Short Term Memory

Long Short Term Memory networks (LSTM) merupakan sebuah evolusi dari arsitektur RNN, dimana pertama kali diperkenalkan oleh Hochreiter & Schmidhuber (1997). Hingga penelitian ini dilakukan banyak para peneliti yang terus mengembangkan arsitektur LSTM di berbagai bidang seperti dalam bidang speech recognition dan forecasting[8].

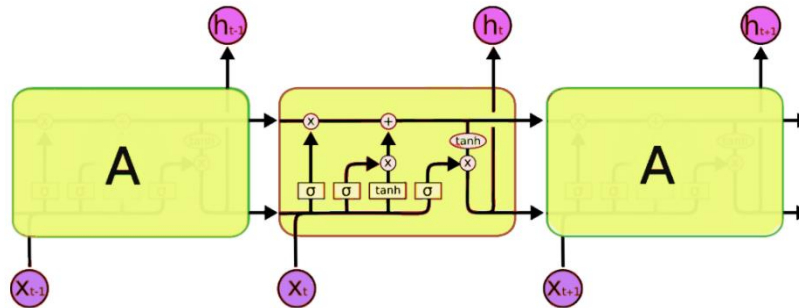


Gambar 2. Memori pada RNN[8]

Pada Gambar 2 menjelaskan RNN memiliki kekurangan yang bisa dilihat pada inputan x_0, x_1 memiliki rentang informasi yang sangat besar dengan x_t, x_{t+1} sehingga ketika h_{t+1} memerlukan informasi yang relevan dengan x_0, x_1 RNN tidak dapat untuk belajar menghubungkan informasi karena memori lama yang tersimpan akan semakin tidak berguna dengan seiringnya waktu berjalan karena tertimpa atau tergantikan dengan memori baru, permasalahan ini ditemukan oleh Bengio, et al. (1994). Berbeda dengan RNN, LSTM tidak memiliki kekurangan tersebut karena LSTM dapat mengatur memori pada setiap masukannya dengan menggunakan *memory cells* dan *gate units*.

Memory cells dan gate units

Seperti yang sudah dijelaskan pada sub bab sebelumnya dimana LSTM memiliki *memory cells* dan *gate units* pada setiap jaringan atau *neurons* nyayang berfungsi sebagai pengatur memori dalam setiap *neurons* atau jaringan[8].



Gambar 3. Memory cells pada LSTM[8]

Pada Gambar 3 menjelaskan bagaimana alur kerja *memory cells* pada setiap *neurons* LSTM bekerja. Terdapat empat proses fungsi aktivasi pada setiap masukan pada *neurons* yang selanjutnya disebut sebagai *gates units*. *Gates units* tersebut ialah *forget gates*, *input gates*, *cell gates*, dan *output gates*.

Pada *forget gates* informasi pada setiap data masukan akan diolah dan dipilih data mana saja yang akan disimpan atau dibuang pada *memory cells*. Fungsi aktivasi yang digunakan pada *forget gates* ini adalah fungsi aktivasi sigmoid. Dimana hasil keluarannya antara 0 dan 1. Jika keluarannya adalah 1 maka semua data akan disimpan dan sebaliknya jika keluarannya 0 maka semua data akan dibuang. Dengan rumus sebagai berikut :

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

Pada *input gates* terdapat dua gates yang akan dilaksanakan, pertama akan diputuskan nilai mana yang akan diperbarui menggunakan fungsi aktivasi sigmoid. Selanjutnya fungsi aktivasi tanh akan membuat vector nilai baru yang akan disimpan pada *memory cell*. Dengan rumus sebagai berikut :

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\check{c}_t = \tanh (W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

Pada *cell gates* akan mengganti nilai pada *memory cell* sebelumnya dengan nilai *memory cell* yang baru. Dimana nilai ini didapatkan dari menggabungkan nilai yang terdapat pada *forget gate* dan *input gate*. Dengan rumus sebagai berikut :

$$c_t = (f_t * c_{t-1} + i_t * \check{c}_t) \tag{5}$$

Pada *output gates* terdapat dua gates yang akan dilaksanakan, pertama akan diputuskan nilai pada bagian *memory cell* mana yang akan dikeluarkan dengan menggunakan fungsi aktivasi sigmoid. Selanjutnya akan ditempatkan nilai pada *memory cell* dengan menggunakan fungsi aktivasi tanh. Terakhir kedua *gates* tersebut di dikalikan sehingga menghasilkan nilai yang akan dikeluarkan. Dengan rumus sebagai berikut :

$$o_t = \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t \tanh (c_t) \tag{7}$$

Sentistrength

Sentistrength merupakan alat atau program yang di kembangkan oleh *Cyberemotion*. Kegunaan dari *sentristrength* sendiri adalah untuk mendeteksi kekuatan sentimen dengan menggunakan pendekatan leksikon atau kamus[9]. *Sentistrength* akan menghasilkan *Positive value* atau *Negative value* dengan *value* atau nilai yang dimulai dari 1 sampai 5. Hanya *value* atau nilai terbesar dari masing-masing *emotion* yang ditampilkan sebagai *final value* atau nilai akhir. Setelah mendapatkan nilai akhir maka dapat dilakukan keputusan akhir yang didasarkan pada.

if positive value > negative value ;

positive sentiment

if positive value < negative value ;

negative sentiment

if positive value = negative value ;

neutral sentiment

penggunaan *sentistrength* ini dapat menggunakan berbagai macam bahasa, namun untuk mendapatkan akurasi yang lebih harus menggunakan *English* sebagai kamus utamanya. Oleh karena itu pada penelitian kali ini, dokumen atau data akan diubah terlebih dahulu ke bahasa inggris sebelum menggunakan *sentistrength* ini. Karena data sudah dalam bentuk *excel* maka, proses *translasi* akan menggunakan fitur *translate* yang ada dalam *excel*.

Term Frequency Inverse Document Frequency

TF (Term Frequency) adalah frekuensi dari kemunculan sebuah term dalam dokumen yang bersangkutan. Semakin besar jumlah kemunculan suatu term (TF tinggi) dalam dokumen, semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar. IDF (Inverse Document Frequency) merupakan sebuah perhitungan dari bagaimana term didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Pembobotan ini bertujuan untuk mengubah kata menjadi vektor. Vektor yang dimaksud dalam tahap ini adalah memberikan bentuk matriks pada kata dasar. TF-IDF dapat ditulis dengan persamaan:

$$TF \times IDF(t, c) = TF(d, t) \times \log_2 \frac{N}{A+C} \quad (8)$$

Dimana :

TF (d, t) : frekuensi kemunculan kata t pada dokumen d

N : jumlah dokumen

A : banyaknya dokumen pada kelas c yang memuat kata t

C : banyaknya dokumen yang terdapat pada kelas c namun tidak memuat term t

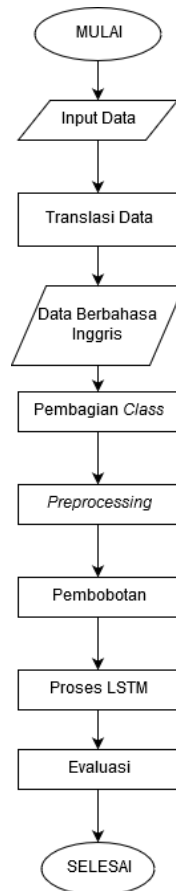
c : kelas kategori

t : kata

3. Perancangan Sistem

Pada penelitian ini, akan dilakukan proses perancangan sistem yang dibangun untuk mengklasifikasikan data teks menggunakan proses LSTM sebagai awal mula metode RNN. Dalam hal ini, proses LSTM membutuhkan berbagai macam proses sebelum sepenuhnya dapat digunakan pada proses RNN. Berikut proses analisis dengan RNN. Pemrosesan RNN dengan model memproses banyak input namun hanya memiliki satu output sangat membantu dalam analisis sentimen. LSTM adalah tipe RNN. Prediksi secara berurutan di RNN, dan lapisan tersembunyi dari satu prediksi adalah yang tersembunyi lapisan prediksi berikutnya. Ini akan menetapkan memori ke jaringan. Hasil dari prediksi sebelumnya dapat ditingkatkan prediksi masa depan. LSTM memberikan RNN aspek ekstra itu memberikan kendali yang lebih baik atas memori. Aspek-aspek

ini mengontrol berapa banyak masukan yang saat ini penting dalam menciptakan memori baru, dan berapa banyak memori sebelumnya yang penting dalam menciptakan memori baru, dan bagian apa dari memori yang penting dalam menghasilkan output. Berikut adalah flowchart dari proses analisis.



Gambar 4. Rancangan Sistem yang akan dibuat

Pengumpulan data

Penelitian ini dilakukan untuk kuesioner yang berbasis bahasa Indonesia. Data yang dikumpulkan berasal dari website akademik Telkom University i-Gracias. Data yang dikumpulkan adalah data teks acak.

Tabel 1. Contoh isian kuesioner

<p>Berikan saran/masukan Saudara/i terkait dengan implementasi layanan di Universitas Telkom dalam bidang Layanan Akademik</p>	<p>Bapak/ ibu maaf sebelumnya saya adalah alumni SMK TELKOM Makassar dan sampai sekarang beasiswa saya belum masuk. Mohon Bapak/ Ibu mempertimbangkan hal ini karena status saya sebagai yatim piatu dan saya belum bekerja sedangkan uang yang saya masukkan belum terpotong. Mohon pertimbangannya. Terima kasih</p>
<p>Berikan saran/masukan Saudara/i terkait dengan implementasi layanan di</p>	<p>Semoga pihak layanan akademik dapat mempertahankan kinerjanya yang bernilai positif</p>

Universitas Telkom dalam bidang Layanan Akademik	dan meningkatkan kinerjanya di masa depan. Terima kasih.
Berikan saran/masukan Saudara/i terkait dengan implementasi layanan di Universitas Telkom dalam bidang Layanan Akademik	Sudah cukup baik

Tabel diatas merupakan salah satu contoh dari isian kuesioner pada website akademik Telkom University i-Gracias. Dalam tabel tersebut terdapat opini atau masukan dari mahasiswa tentang layanan akademik. Dengan mengolah data dari opini atau masukan tersebut kita dapat mengetahui apakah layan akademik Telkom University cenderung dengan opini positif atau negatif.

Pembagian Data

Pembagian data dilakukan untuk membagi data menjadi data training dan data testing. Hal ini dilakukan agar dapat dilakukannya suatu perbandingan akurasi pada data yang sedang diolah. Pembagian data juga bertujuan untuk mempermudah pengklasifikasian pada data, sehingga data yang dihasilkan bisa lebih tepat. Pada penelitian ini penulis membuat skenario pembagian data, yaitu: data latih dan data uji (70:30), (80:20), dan (50:50).

Pre-processing

Pada tahap ini, *data training* dan *data testing* akan dilakukan *preprocessing data* untuk menghilangkan data yang tidak sempurna. Beberapa tahapan diantaranya ialah *case folding*, *tokenizing*, *filtering* dan *stemming*.

- a. *Case folding*, yaitu mengubah seluruh huruf kapital menjadi huruf kecil.
- b. *Tokenizing*, yaitu mengubah kalimat menjadi kumpulan satu kata.
- c. *Filtering*, yaitu menghilangkan *stop word*.
- d. *Stemming*, yaitu mengembalikan kata ke dalam bentuk dasar (kata dasar) dengan menghilangkan aditif yang ada.

Pembobotan TF-IDF

Pada proses ini data *kueisoner* yang telah dikumpulkan sebelumnya akan di proses untuk dilakukan pembobotan yang bertujuan untuk mendapatkan rating pada kata-kata yang didapatkan. Dua metode pembobotan ini akan dicoba untuk mengetahui seberapa berpengaruh kata dari suatu dokumen nantinya. Hasil dari pembobotan berupa banyak kata yang telah diberi bobot.

Proses LSTM

Proses ini merupakan proses paling penting yaitu untuk mengklasifikasikan data yang sudah di proses pada tahap pembobotan sebelumnya, dengan menggunakan salah satu dari metode RNN yaitu *Long Short Term Memory (LSTM)*. Pada tahap ini *data training* akan diinputkan dan mendapatkan hasil atau *output* berupa model prediksi untuk *data testing* digunakan untuk membantu hasil ujicoba penelitian ini

4. Hasil dan Analisis

Pada bagian ini akan dijelaskan bagaimana hasil uji dari sistem yang sudah di buat sebelumnya.

Data set

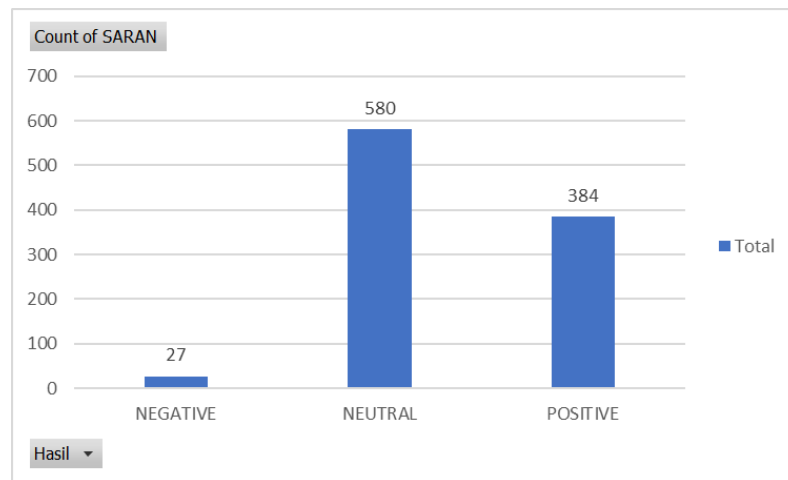
Pada data kuesioner yang didapatkan pada website akademik Telkom University I-Gracias diambil 1.000 opini saran mahasiswa Telkom University untuk dijadikan data latih dan data uji. Sebelum data dibagi menjadi data latih dan data uji perlu dibuatkan *class* untuk setiap opini pada data kuesioner. Pembagian *class* ini memerlukan alat bantu yaitu program yang bernama *sentistrength*. Dengan

menggunakan alat bantu ini sentimen akan mudah didapatkan dan memepermudah pembagian *class*. Berikut merupakan contoh dari hasil penggunaan *sentistrength*.

tabel 2. Hasil pembagian *class* menggunakan *sentistrength*

SARAN	TRANSLATION SUGGETIONS	Positive	Negative	Nilai akhir	Hasil
semoga kedepannya bisa lebih baik lagi	hopefully in the future it could be better	1	-1	0	NEUTRAL
semoga lebih bisa diperbaiki lagi yang masih kurang	may be irreparably still lacking	1	-1	0	NEUTRAL
Tingkatkan	Increase	1	-1	0	NEUTRAL
Y	Y	1	-1	0	NEUTRAL
bagus	nice	2	-1	1	POSITIVE
sudah cukup baik tingkatkan lagi	is good enough to increase again	2	-1	1	POSITIVE
lebih diperbaiki lagi	more irreparably	1	-1	0	NEUTRAL
cukup baik, lebih ditingkatkan lagi	pretty good, much improved	3	-1	2	POSITIVE
belum pernah ngurusin ke LAK	has never ngurusin to LAK	1	-1	0	NEUTRAL

didapatkan hasil total *class* untuk *class positive* dan *class negative*. Berikut merupakan gambar dari hasil total *class* dari penggunaan *sentistrength*.



Gambar 5. Hasil total *class* yang didapatkan

Pada gambar5 dapat dilihat bahwa opini pada dokumen sudah memiliki *class* nya masing-masing. Dengan menggunakan *class* ini tahapan selanjutnya bisa dilakukan.

Hasil preprocessing dan pembobotan

Pada tahap ini, semua opini kueisoner pada dokumen akan dikumpulkan menjadi satu bagian dan setelah itu akan dilkakukan *preprocessing* untuk mengubah data menjadi terstruktur. Proses *preprocessing* yang pertama adalah *Case folding* yaitu mengubah huruf besar menjadi huruf kecil. *Tokenizing* merupakan proses penghapusan karakter seperti titik (.) dan koma (,) lalu, kalimat diuraikan menjadi satuan kata. *Filtering* yaitu pemilihan kata-kata penting setelah proses *tokenizing*. Terakhir adalah *stemming* yaitu proses mengubah kata yang berimbuhan menjadi kata dasar.

Tabel 3. Contoh hasil preprocessing

lebih diperbagus lagi	bagus
sudah baik	baik
bagus	bagus
lebih mendorong mahasiswa untuk kreatif	kreatif

Selanjutnya masuk ke tahap pembobotan atau vektorisasi, semua kata hasil *preprocessing* akan di beri bobot dengan menggunakan perhitungan TF-IDF. Berikut adalah hasil dari pembobotan dengan menggunakan perhitungan TF-IDF.

Tabel 4. Contoh hasil Pembobotan atau vektorisasi TF-IDF

NEUTRAL	depan bisa baik masih ku	1 1 1 0 0 0	22 21 329	15.0246 15
NEUTRAL	depan bisa baik masih ku	0 1 1 1 1 0	22 21 329	15.0246 15
NEUTRAL	depan bisa baik masih ku	0 0 0 0 0 0	22 21 329	15.0246 15

Long Short Term Memory (LSTM)

Setelah memlalui tahap *preprocessing* dan pembobotan akan masuk ke tahap klasifikasi menggunakan metode LSTM. Pengujian ini menggunakan pembagian data dengan ratio (70:30), (80:20) dan (50:50), dengan menggunakan iterasi sebanyak 100. Berikut adalah hasil dari proses yang sudah dilakukan.

Tabel 5. Hasil dari Proses LSTM

pembagian ratio data	akurasi Positive	akurasi negative	akurasi Neutral
70, 30	55,31%	79,68%	82,49%
80, 20	56,70%	79,84%	81,88%
50, 50	56,73%	78,41%	80,79%

Hasil Uji

Dari seluruh percobaan dengan menggunakan beberapa fitur dan percobaan penggunaan rasio data set, didapat hasil yang berbeda-beda. Setiap pembagian ratio data memiliki hasil akurasi terbesar dibandingkan dengan ratio lainnya. Hasil akurasi untk *class positive* terbaik ada pada ratio (50:50) dengan akurasi 56,73%, *class negative* terbaik ada pada ratio (80:20) dengan akurasi 79,84% , dan *class neutral* terbaik ada pada ratio (70:30) dengan akurasi 82,49%.

5. Kesimpulan

Penelitian dengan menggunakan salah satu metode *RNN* yaitu LSTM untuk klasifikasi data kuesioner membuktikan apakah metode ini mampu mengklasifikasikan data kuesioner dengan harapan dapat menjadi salah satu metode lain yang dapat membantu dalam pengklasifikasian data. Pada penelitian ini penggunaan metode LSTM terbukti mamapu mengklasifikasikan data kuesioner dengan akurasi yang cukup baik. Unutk mendapatkan akurasi yang baik diperlukan pembagian ratio data yang tepat, seperti pada akurasi *class positive* dengan menggunakan raito data 50:50. Terdapat beberapa kesalahan dalam penggunaan metode ini, salah satunya pada proses translasi bahasa. Namun hal ini dapat dihindari jika saat proses translasi bahasa lebih diperhatikan lagi struktur kata bahasa indonesia dengan bahasa inggrisnya. Saran untuk penelitian selanjutnya adalah pada saat pembagian *class* menggunakan *sentistrength* pasitkan data sudah ditranslasi dengan menggunakan struktur kalimat yang sesuai agar pembagian *class* menjadi akurat. Lalu pembagian ratio data yang tepat juga perlu diperhatikan untuk membantu mendapatkan hasil yang maksimal.

Daftar pustaka

- [1] S. K. Lidya, O. S. Sitompul, and S. Efendi, "Sentiment Analysis Pada Teks Bahasa Indonesia Menggunakan Support Vector Machine (SVM) dan K-Nearest Neighbor(K-NN)," *Semin. Nas. Teknol. dan Komun. 2015*, vol. 2015, no. Sentika, pp. 1–8, 2015.
- [2] B. Liu, "Liu - 2010 - Sentiment Analysis and Subjectivity.pdf," pp. 1–38, 2010.
- [3] I. F. Rozi, S. H. Pramono, and E. A. Dahlan, "Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi," *Electr. Power, Electron. Commun. Control. Informatics Semin.*, vol. 6, no. 1, pp. 37–43, 2012.
- [4] Y. Ando *et al.*, "Presence of autoantibody against attr Val30Met after sequential liver transplantation," *Transplantation*, vol. 73, no. 5, pp. 751–755, 2002.
- [5] A. Ahmad, "Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning," *Teknol. Indones.*, no. October, pp. 1–3, 2017.
- [6] A. Rios and R. Kavuluru, "Convolutional neural networks for biomedical text classification," *Proc. 6th ACM Conf. Bioinformatics, Comput. Biol. Heal. Informatics - BCB '15*, pp. 258–267, 2015.
- [7] P. Liu, X. Qiu, and H. Xuanjing, "Recurrent neural network for text classification with multi-task learning," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2016-Janua, pp. 2873–2879, 2016.
- [8] M. Wildan, P. Aldi, and A. Aditsania, "Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin," vol. 5, no. 2, pp. 3548–3555, 2018.
- [9] R. Sianipar and E. B. Setiawan, "PENDETEKSIAN KEKUATAN SENTIMEN PADA TEKS TWEET BERBAHASA INDONESIA MENGGUNAKAN SENTISTRENGTH," vol. 2, no. 3, pp. 7922–7928, 2015.
- [10] <https://indoml.files.wordpress.com/2018/04/intuisi-rnn-translasi-bahasa2.jpg?w=736>