Deteksi Ujaran Kebencian pada Komentar Instagram dalam Bahasa Indonesia Menggunakan Metode Recurrent Neural Network

Hengky [1] Yulius Denny Prabowo [2]

Informatika, Fakultas Industri Kreatif Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav 22, Jakarta 13210
[1]*Email: okytriputra@gmail.com*
[2]*Email: yulius.prabowo@kalbis.ac.id*

Abstract: *Hate speech is any form of communication that are provoking and bringing down a group or individual. The objective of this study is to create a model that can detect hate speech on Instagram comments in Indonesian language. The method used in this study is Recurrent Neural Network. This study produces a model that can detect hate speech with an accuracy 81%.*
Keywords: *nlp, hate speech, recurrent neural network, lstm, instagram*

Abstrak: *Ujaran kebencian adalah segala bentuk komunikasi yang bersifat memprovokasi dan menjatuhkan suatu kelompok atau individu. Penelitian ini bertujuan untuk membuat sebuah model yang dapat mendeteksi ujaran kebencian pada komentar Instagram dalam bahasa Indonesia. Metode yang digunakan dalam penelitian ini adalah Recurrent Neural Network. Peneltian ini menghasilkan sebuah model yang dapat mendeteksi ujaran kebencian dengan akurasi 81%.*
Kata Kunci: *nlp, ujaran kebencian, recurren neural network, lstm, instagram*

I. PENDAHULUAN

Media sosial adalah sebuah media atau wadah dalam bentuk virtual yang terkoneksi dengan jaringan internet sehingga orang-orang dapat melakukan aktivitas sosial di dalamnya [1]. Aktivitas sosial yang dimaksud berupa membuat, membagikan, dan bertukar informasi. Macam-macam media sosial yaitu Instagram, Facebook, Twitter, dan masih banyak lagi. Saat ini media sosial digunakan oleh banyak orang.

Pengguna sosial media di dunia sampai pada bulan Juli 2019 telah mencapai lebih dari 3,5 miliar orang. Jumlah pengguna media sosial seperti Instagram dan Facebook di Indonesia merupakan peringkat ke-4 di dunia [2]. Hal ini menyebabkan banyak informasi yang tersebar dan dilihat oleh pengguna sosial media tersebut. Penggunaan sosial media ini terbilang bebas karena kita bisa memposting apa saja ke sosial media tersebut tak terkecuali ujaran kebencian.

Ujaran kebencian adalah segala komunikasi yang menunjukan atau mempengaruhi orang lain untuk membenci terhadap suatu golongan, ras, fisik, gender, sifat, agama, dan lain-lain [3]. Ujaran kebencian menyebabkan dampak yang buruk. Dampak buruk yang dapat terjadi adalah timbulnya provokasi yang menyebabkan perpecahan. Selain dampak perpecahan, ujaran kebencian dapat menyebabkan gangguan psikologis pada seseorang.

Perkembangan teknologi serta sosial media menyebabkan kita dapat lebih mudah menyebarkan ujaran kebencian. Banyaknya pengguna sosial media di Indonesia membuat kasus mengenai ujaran kebencian harus menjadi sorotan. Pada tahun 2018 terdapat 122 orang tertangkap akibat ujaran kebencian di media sosial oleh kepolisian [4]. Ini arti

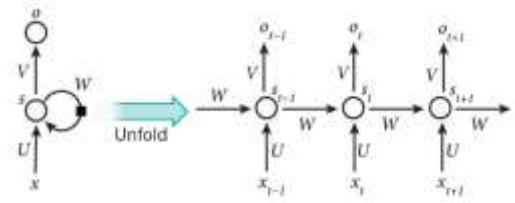
nya terdapat 10 kasus bahkan lebih dalam 1 bulan. Kasus ini mungkin saja akan bertambah tiap tahunnya apalagi 2019 merupakan tahun pemilu.

Dalam penelitian ini, data yang digunakan bersumber dari komentar di Instagram yang dimodelkan menggunakan metode Recurrent Neural Network (RNN). Data komentar diambil dari komentar di Instagram menggunakan Instagram Scraper yang merupakan aplikasi command line yang diambil dari github. Data yang diambil akan diberikan label pembeda secara manual antara komentar yang berisi kebencian dengan yang tidak. Hasil dari penelitian ini adalah membuat sebuah aplikasi yang bertujuan mempelajari pola dari ujaran kebencian dan mendeteksi komentar lain yang berisi ujaran kebencian berdasarkan pola tersebut.

II. METODE PENELITIAN

A. Reccurent Neural Network (RNN)

Recurrent Neural Network (RNN) adalah jenis arsitektur pada neural network yang pemrosesannya dilakukan berulang-ulang untuk memproses masukan yang bersifat sequences [5]. RNN merupakan model yang dapat menyelesaikan permasalahan NLP dengan baik [6]. Model RNN berfungsi untuk mengolah data yang berbentuk sequences [6]. Berbeda dengan neural network biasa yang setiap input dan output bersifat independen pada masing-masing neuron [6]. Model ini disebut reccurent karena pemrosesannya yang bersifat berulang dan output yang dihasilkan bergantung pada pemrosesan sebelumnya [6]. Tampilan arsitektur model ini dapat dilihat pada Gambar 1



Gambar 1 Arsitektur RNN

Di dalam RNN terjadi berbagai perhitungan. Misalkan, t adalah langkah waktu yang dilalui model. Langkah pertama yang dilakukan adalah menghitung hidden state S_t yang berasal dari input X_t dan hidden state sebelumnya yaitu S_{t-1} [5]. X_t dan S_{t-1} dikalikan dengan parameter U dan W yang kemudian dimasukan kedalam fungsi aktifasi tanh [5].

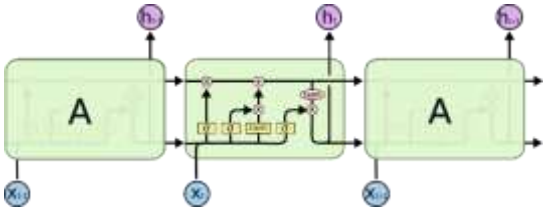
$$s_t = \tanh(U \,.\, x_t + W \,.\, s_{t-1})$$

S_t yang dihasilkan digunakan untuk input tambahan pada state berikutnya serta output pada layer t [5]. Untuk menghasilkan output t atau Y_t, S_t dikalikan menggunakan parameter V dan dimasukan kedalam fungsi aktifasi softmax [5].

$$\hat{y}_t = softmax(V \,.\, s_t)$$

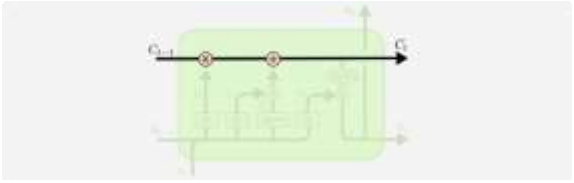
B. Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) merupakan salah satu jenis RNN yang dimodifikasi pada memory cell sehingga dapat menyimpan informasi dalam jangka waktu yang panjang [7]. LSTM menjadi jawaban atas permasalahan *vanishing gradient* pada RNN saat mengolah data *sequence* yang panjang [7]. LSTM didesain untuk mengatasi *vanishing gradient* dengan menggunakan mekanisme gerbang [8]. Pada arsitektur LSTM terdapat 3 gerbang yaitu gerbang *input*, gerbang *forget*, dan gerbang *output* [7].



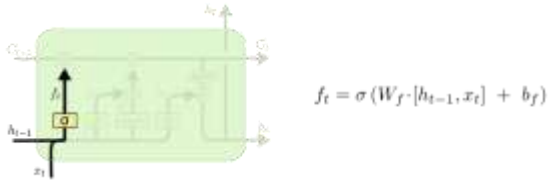
Gambar 2 Arsitektur LSTM

Inti dari sebuah LSTM terdapat pada *Cell state* dan konsep gerbang [9]. *Cell state* dapat diumpamakan sebagai jalur yang menyampaikan informasi antar rantai *sequences* [9]. *Cell state* membawa informasi melewati setiap sel sehingga dapat dikatakan sebagai memori [9]. Ketiga gerbang yang berada pada LSTM berfungsi untuk menjaga dan mengontrol *cell state* [10].



Gambar 3 *Cell state*

Langkah pertama yang dilakukan LSTM adalah menentukan informasi yang tidak diperlukan oleh *cell state* [10]. Keputusan yang diambil untuk menyimpan atau membuang informasi dibuat oleh sebuah *layer* sigmoid yang bernama *forget gate* [10].

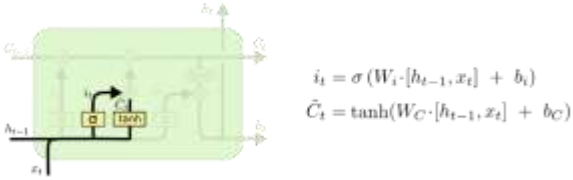


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Gambar 4 *Forget gate*

Informasi dari *hidden state* (h_{t-1}) sebelumnya dan input saat ini (x_t) dimasukan kedalam fungsi sigmoid [9]. Hasil dari operasi ini merupakan nilai 0 atau 1 [9]. Nilai 0 berarti lupakan informasi sedangkan nilai 1 artinya simpan [9].

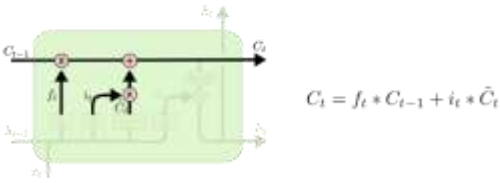
Input gate digunakan untuk memperbarui *cell state* [10]. *Hidden state* (h_{t-1}) sebelumnya dan input sekarang (x_t) dimasukan kedalam fungsi sigmoid [9]. Hasil dari perhitungan tersebut akan menghasilkan nilai 0 atau 1 [9]. Nilai ini berfungsi untuk menentukan informasi yang akan diupdate pada *cell state* [24]. Nilai 0 berarti tidak penting sedangkan nilai 1 berarti penting [9]. *Hidden state* (h_{t-1}) dan input yang sekarang (x_t) dimasukan kedalam fungsi tanh untuk menghasilkan sebuah vector nilai *candidate* yang baru (C_t).



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Gambar 5 *Input gate*

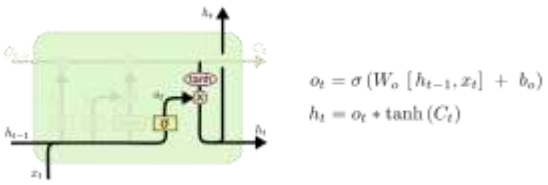
Tahapan berikutnya adalah memperbarui nilai dari *cell state*. Cell state yang lama (C_{t-1}) dikalikan dengan f_t yang dihasilkan oleh *forget gate* [10]. Kemudian ditambahkan dengan hasil perkalian dari i_t dan C_t. Hasil dari operasi ini adalah nilai *cell state* yang baru yang akan dimasukan kedalam sel selanjutnya.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Gambar 6 Perhitungan *cell state*

Output gate berfungsi untuk menentukan nilai *hidden state* yang akan diteruskan ke sel selanjutnya [9]. *Hidden state* mengandung informasi yang ada pada sel sebelumnya [9]. Pertama *Hidden state* pada sel sebelum (h_{t-1}) dan input pada sel sekarang (X_t) dimasukan kedalam fungsi sigmoid [9]. *Cell state* yang telah diperbarui dimasukan kedalam fungsi tanh [9]. Hasil dari fungsi sigmoid dan tanh kemudian dikalikan dan menghasilkan *hidden state* (h_t) yang akan diteruskan ke sel selanjutnya [9].



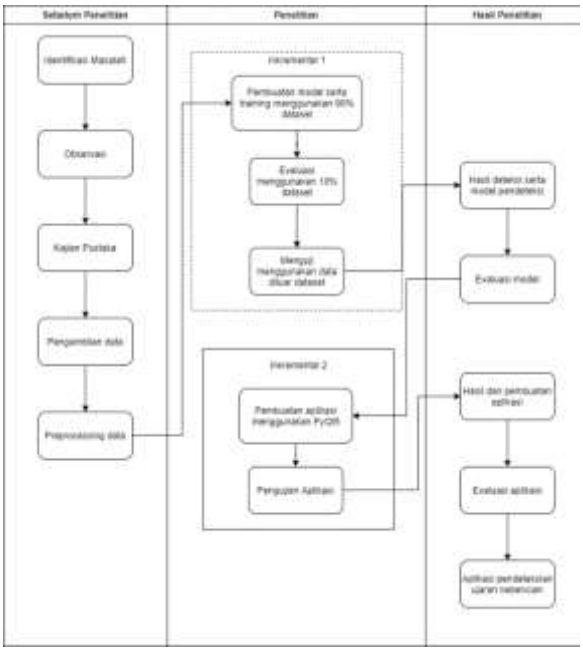
$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Gambar 7 *Output gate*

C. Proses Penelitian

Pada penelitian ini, metode yang digunakan oleh peneliti dalam pengembangan aplikasi deteksi ujaran kebencian merupakan metode inkremental. Pada pengembangan aplikasi ini, terdapat 2

inkremental yaitu Inkremental 1 dan Inkremental 2. Proses dari penelitian ini dapat dilihat pada gambar 8.



Gambar 8 Proses penelitian

Penelitian ini bertujuan untuk membuat sebuah aplikasi yang dapat digunakan untuk mendeteksi ujaran kebencian dalam bahasa Indonesia. Data yang digunakan dalam penelitian ini berasal dari komentar instagram yang diambil menggunakan aplikasi yang bernama *Instagram Scraper*. Komentar yang diambil merupakan komentar acak yang berada pada sebuah *posting*-an yang dapat menjadi sumber ujaran kebencian.

Observasi yang dilakukan peneliti adalah menemukan kasus yang memiliki potensi sumber dari ujaran kebencian. Pada saat melakukan observasi ada beberapa kasus yang menurut peneliti dapat menjadi sumber ujaran kebencian di *instagram*. Pada saat melakukan observasi, terdapat beberapa kasus yang dapat menjadi sumber ujaran kebencian. Kasus yang dapat menyebabkan kasus ujaran kebencian antara lain banjir pada bulan januari 2020, penyebaran virus corona, dan kasus selebgram yang mengambil *handsanitizer* Ripanzul. Dari berbagai kasus yang peneliti telah sebutkan, peneliti memilih kasus Ripanzul untuk pengambilan data komentar. Kasus ini dipilih karena kasus ini cukup viral dan banyak orang-orang yang tidak menyukai dan menyerang akun instagramnya.

Inkremental 1 berisi proses pembuatan model serta proses *training* model menggunakan 90% data dan *testing* menggunakan 10% dari dataset. Keluaran dari Inkremental 1 adalah model yang telah dilatih serta akurasi dari model tersebut. Pada Inkremental 2, berisi proses pembuatan tampilan aplikasi serta fungsi-fungsi pada aplikasi tersebut. Hasil dari inkremental 2 adalah aplikasi deteksi ujaran kebencian. Output dari aplikasi ini adalah file format csv yang berisi username, komentar, dan label yang berisi angka 0-1. Semakin mendekati 1 artinya komentar tersebut semakin mengandung ujaran kebencian.

1. Pengumpulan Data

Data yang digunakan dalam penelitian ini diambil menggunakan sebuah aplikasi. Aplikasi teersebut adalah *Instagram Scraper*. Aplikasi ini dapat digunakan menggunakan terminal pada *virtual environment* atau cmd pada windows. Aplikasi ini juga memerlukan python yang terinstall pada pc.

Aplikasi akan secara otomatis membuat folder yang berisi media *post* seperti foto dan video serta file .json yang berisi komentar. Dari file .json tersebut, peneliti hanya membutuhkan data *username* dan data komentar. Untuk mengambil data tersebut, peneliti membuat sebuah algoritma yang berfungsi untuk mengambil data *username* dan komentar serta menjadikan data tersebut menjadi 1 dataframe.

2. Preprocessing

Setelah melakukan penarikan data dari file berformat .json menjadi .csv, peneliti melakukan pelabelan pada tiap komentar. Label terdiri dari angka 0 dan 1. Angka 0 digunakan untuk komentar yang tidak mengandung ujaran kebencian dan angka 1 digunakan untuk komentar yang mengandung ujaran kebencian. Komentar yang diberikan label 1 adalah komentar yang

memiliki kata-kata kasar, menyerang, memprovokasi. Kata-kata tersebut dapat berupa anjing, mati aja, bego, dan lain-lain. Setelah tahap pelabelan, peneliti mendapatkan jumlah komentar yang dikumpulkan adalah 1127.

Setelah proses pelabelan, peneliti melakukaan proses *preprocessing*. Tahapan *preprocessing* yang dilakukan antara lain *case folding* , menghilangkan emoji, menghilangkan tanda baca, dan menghilangkan *stopwords*.

3. Inkremental 1

Sebelum membuat model, peneliti membagi dataset menjadi data training dan data testing. Peneliti membagi dataset menjadi data training sebesar 90% dan data testing sebesar 10%.

Setelah melakukan *train test split*, peneliti membuat sebuah model *tokenizer* untuk melakukan tokenisasi pada data. Tokenisasi adalah proses membuat data menjadi token atau kata perkata. Peneliti membuat model *tokenizer* menggunakan *library keras.* Data yang digunakan untuk *training* model *tokenizer* adalah data *training*. Hal ini dilakukan supaya model hanya mengenali data yang ada pada data training. Selain mengubah data menjadi token, *tokenizer* juga memberikan tiap kata sebuah index. Index yang diberikan berdasarkan frekuensi kata pada data training.

Data komentar kemudian diubah menjadi *sequences*. *Sequences* berasal dari susunan *index* yang dihasilkan tokenizer dan mewakili kata pada tiap komentar. *Sequence* berupa matriks dengan 1 dimensi. Setiap *sequences* yang telah dibuat memiliki ukuran yang berbeda-beda. Untuk memproses data kedalam model, ukuran dari setiap *sequences* harus disamakan. Menyamakan ukuran *sequences* dapat menggunakan metode *padding*. *Padding* adalah metode yang digunakan untuk menyamakan ukuran setiap *sequences*.

Model yang dibuat adalah model dengan menggunakan metode RNN. Model memiliki beberapa *layer* dan *node*.

Layer (type)	Output Shape	Param #
embedding_10 (Embedding)	(None, 85, 64)	146816
dropout_9 (Dropout)	(None, 85, 64)	0
lstm_10 (LSTM)	(None, 32)	12416
dense_18 (Dense)	(None, 10)	330
dense_19 (Dense)	(None, 1)	11

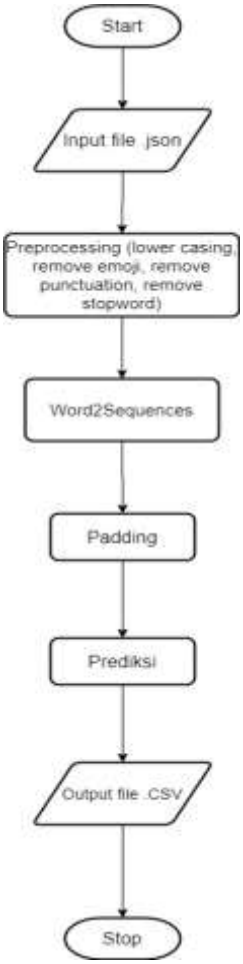
Total params: 159,573
Trainable params: 159,573
Non-trainable params: 0

Gambar 9 Model

Gambar 9 merupakan desain dari model deteksi ujaran kebencian yang akan digunakan. Model memiliki beberapa jenis *layer*. Jenis-jenis *layer* yang digunakan adalah *layer embedding*, *dropout*, *LSTM*, dan *dense.* Setiap layer memiliki node. Jumlah node bervariasi. Banyaknya node ditentukan berdasarkan dimensi input dan output yang diharapkan.

4. Inkremental 2

Pada Inkremental 2, peneliti membuat tampilan antarmuka serta fungsi aplikasi deteksi ujaran kebencian. Dalam pembuatan aplikasi ini, peneliti menggunakan *library PyQt5*. Pada pembuatan tampilan antarmuka aplikasi, peneliti menggunakan aplikasi *QtDesigner*. Aplikasi digunakan untuk membuat desain tampilan antarmuka aplikasi. Keluaran dari aplikasi ini adalah file yang berformat '.ui'. File '.ui' akan diekspor menjadi format '.py' menggunakan *Pyuic5*. Hal ini dilakukan untuk memberikan fungsi.



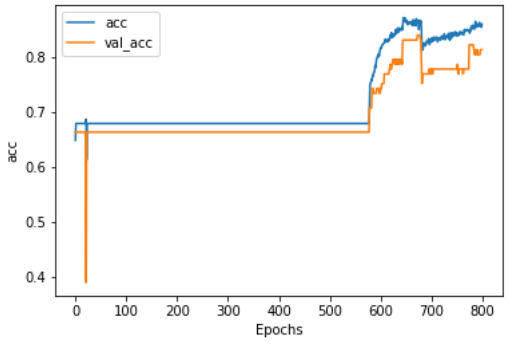
Gambar 10 Diagram alur aplikasi deteksi ujaran kebencian

Gambar 10 merupakan diagram alur dari jalannya aplikasi deteksi ujaran kebencian. Aplikasi ini memerlukan *input* berupa *file* berformat '.json'. Setelah itu, aplikasi akan memproses data serta memprediksi data menggunakan model yang sudah dibuat. Aplikasi ini akan memberikan output sebuah file berformat '.csv'.

III. HASIL DAN PEMBAHASAN

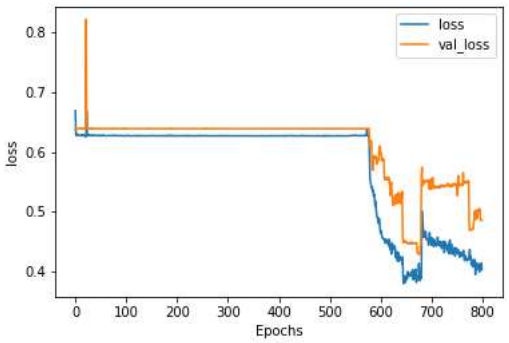
A. Inkremental 1

Peneliti melakukan training dengan menggunakan data yang telah dibagi menjadi 90% untuk data latih dan 10% untuk data uji. Hasil yang didapatkan oleh peneliti adalah akurasi yang mencapai 86% pada data latih dan 81.42% pada data uji. Gambar 11 merupakan grafik perbandingan antara akurasi pada data latih dan data uji.



Gambar 11 Perbandingan akurasi pada data uji dan latih

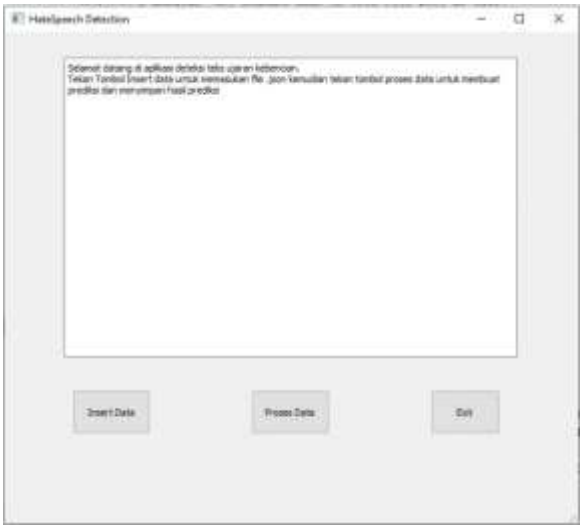
Sedangkan pada loss, loss yang didapatkan pada data latih adalah 0.4030 dan loss pada data uji adalah 0.4856. Gambar 4.8 adalah perbandingan antara loss pada data latih dan uji.



Gambar 12 Perbandingan loss pada data uji dan latih

B. Inkremental 2

Pada tahapan Inkremental II, peneliti melakukan pembuatan tampilan aplikasi deteksi ujaran kebencian. Tampilan ini dibuat menggunakan *library python* yaitu PyQt5 dan dibantu dengan aplikasi QtDesigner. Tampilan dari aplikasi deteksi ujaran kebencian ini dapat dilihat pada gambar 13



Gambar 13 Tampilan aplikasi deteksi ujaran kebencian

Keluaran dari aplikasi ini merupakan sebuah file berekstensi '.csv' yang dapat dibuka menggunakan excel atau python. Ini dari file ini adalah data komentar, *username*, serta hasil prediksi dari aplikasi deteksi ujaran kebencian. Keluaran dari aplikasi deteksi ujaran kebencian dapat dilihat pada gambar

Gambar 14 Keluaran aplikasi ujaran kebencian

C. Pengujian Aplikasi

Pada inkremental II, hasil yang diperoleh adalah tampilan serta fungsi-fungsi dari aplikasi deteksi ujaran kebencian. Peneliti menguji aplikasi dengan menggunakan metode *blackbox*. Berdasarkan pengujian yang telah dilakukan oleh peneliti, setiap tombol dan fungsi dapat berjalan dengan baik.

Tabel 1 Hasil pengujian *blackbox*

No	Pengujian	Hasil yang diharapkan	Hasil yang didapatkan	Keterangan
1	Tombol Insert	Menampilkan jendela untuk pengambilan data dengan format .json kemudian menambahkan tulisan pada *dialog box* data json 'direktori' telah berhasil dimasukan.	Muncul jendela pengambilan data berformat json kemudian menambahkan teks pada *dialog box*	Berhasil
2	Tombol Proses data	Menambahkan kalimat pada dialog box yang menyatakan data telah diproses kemudian memunculkan jendela yang berfungsi untuk menunjukan direktori penyimpanan *file* .csv	Dialog box menambahkan kalimat kemudian aplikasi mengeluarkan jendela untuk menyimpan data dan *file* .csv berada di direktori yang sesuai dengan yang telah ditentukan	Berhasil
3	Tombol Keluar	Menutup aplikasi	Aplikasi tertutup	Berhasil

IV. SIMPULAN

Berdasarkan penelitian yang telah dilakukan, peneliti dapat menyimpulkan beberapa kesimpulan antara lain :

- Metode RNN dapat digunakan untuk melakukan deteksi ujaran kebencian pada teks komentar *Instagram* berbahasa Indonesia.
- Hasil training menggunakan model RNN pada *dataset* yang telah dibuat mendapatkan akurasi sebesar 86% pada data latih dan 81% pada data uji. Akurasi ini menunjukan seberapa dekat angka hasil prediksi model dengan hasil yang diinginkan.
- Loss yang didapatkan pada saat melatih model adalah 0.4030 pada data latih dan 0.4856 pada data uji. Loss ini menunjukan kinerja model, semakin kecil nilai loss semakin baik model.

- Pada model yang dibuat masih ada beberapa kalimat yang seharusnya terdeteksi ujaran kebencian tetapi tidak begitu juga sebaliknya. Hal ini disebabkan data yang kurang banyak dan bervariasi sehingga menyebabkan masih ada beberapa kalimat yang tidak terdeteksi.
- Aplikasi deteksi ujaran kebencian berinput file berekstensi '.json' dan menghasilkan file yang berekstensi '.csv'

DAFTAR RUJUKAN

[1] Ambar, "20 Pengertian Media Sosial Menurut Para Ahli," 8 Juni 2017. [Online]. Available: https://pakarkomunikasi.com/pengertian-media-sosial-menurut-para-ahli. [Accessed 11 Maret 2020].

[2] A. S. Wardani, "Jumlah Pengguna Instagram dan Facebook Indonesia Terbesar ke-4 di Dunia," 26 Juni 2019. [Online]. Available: https://www.liputan6.com/tekno/read/3998624/jumlah-pengguna-instagram-dan-facebook-indonesia-terbesar-ke-4-di-dunia. [Accessed 11 Maret 2020].

[3] R. M. M. I. F. Y. E. Ika Alfina, "Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study," *ICACSIS 2017*, vol. 1, no. 4, p. 1, 2017.

[4] A. Gabrillin, "Selama 2018, Polisi Tangkap 122 Orang Terkait Ujaran Kebencian di Medsos," kompas, 15 02 2019. [Online]. Available: https://nasional.kompas.com/read/2019/02/15/15471281/selama-2018-polisi-tangkap-122-orang-terkait-ujaran-kebencian-di-medsos. [Accessed 31 10 2019].

[5] "Pengenalan Recurrent Neural Network (RNN) – Bagian 1," indoml, [Online]. Available: https://indoml.com/2018/04/04/pengenalan-rnn-bag-1/. [Accessed 16 June 2020].

[6] D. Britz, "Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs," 17 September 2015. [Online]. Available: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/. [Accessed 16 June 2020].

[7] A. S. Girsang, "LONG SHORT TERM MEMORY (LSTM)," Bina Nusantara, 2 December 2019. [Online]. Available: https://mti.binus.ac.id/2019/12/02/long-short-term-memory-lstm/. [Accessed 17 June 2020].

[8] R. Ashrovy, "Recurrent Neural Network — Part Four (END)," 20 October 2017. [Online]. Available: https://medium.com/@ashrovy/recurrent-neural-network-part-4-d371474b8fa9. [Accessed 17 June 2020].

[9] M. Phi, "Illustrated Guide to LSTM's and GRU's: A step by step explanation," 25 September 2018. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21. [Accessed 17 June 2020].

[10] "Understanding LSTM Networks," 27 August 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed 17 June 2020].