

IMPLEMENTASI DEEP LEARNING UNTUK DETEKSI EKSPRESI EMOSI PADA TWITTER

Adinda Dwi L.¹, Budi Harijanto², Faisal Rahutomo³

^{1,2,3}Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang
¹larasatidinda@gmail.com, ²budi.hijet@gmail.com, ³faisal@polinema.ac.id

Abstrak—Twitter menjadi salah satu media berjejaring yang digunakan oleh peneliti pada bidang penelitian media sosial. Banyak hal yang bisa diteliti dari kumpulan tweet. Salah satu yang populer dilakukan adalah menganalisa emosi dari tweet. Emosi dapat dideteksi melalui ekspresi suara, wajah, maupun tulisan. Dalam penelitian ini, dilakukan deteksi ekspresi emosi pada konteks tulisan atau teks dari Twitter. Memahami emosi dengan konteks tulisan cukup sulit karena ekspresi wajah dan/atau suara tidak diketahui. Penelitian untuk menganalisa emosi yang dilakukan pada Twitter juga masih banyak yang menggunakan metode *machine learning*. Untuk itu penelitian kali ini dilakukan dengan metode *deep learning* dengan menggunakan metode LSTM-GloVe untuk mendeteksi label emosi dari suatu tweet berbahasa Indonesia. Data yang digunakan berjumlah 4.403 dari 5 jenis emosi yaitu *anger*, *love*, *sadness*, *fear*, dan *love* dari penelitian “*Emotion Classification on Indonesian Twitter Dataset*”. Dari hasil uji coba dengan menggunakan metode LSTM-GloVe didapatkan nilai *precision*, *recall*, dan *F1* masing-masing sebesar 33%, 38%, dan 35%. Adapun nilai *loss* dan *accuracy* dari pengujian model tersebut sebesar 1,29 dan 0,5.

Kata kunci—*Deep Learning*, *LSTM*, *GloVe*, *Emosi*, *Twitter*.

I. PENDAHULUAN

Media sosial adalah sekelompok aplikasi berbasis internet yang dibangun berdasarkan kerangka pemikiran ideologi dan teknologi *Web 2.0*, dan memungkinkan terbentuknya kreasi dan pertukaran isi informasi dari pengguna internet [1]. Media sosial menjadi sebuah tren untuk berkomunikasi dengan orang lain. Menurut hasil survey APJII atau Asosiasi Penyelenggara Jasa Internet Indonesia, persentase pengguna internet di Indonesia pada tahun 2018 naik 10,12% dari tahun sebelumnya. Hal ini memiliki arti bahwa sekitar 64,8% penduduk Indonesia atau 171,17 juta jiwa dari total populasi penduduk Indonesia menggunakan internet. Penggunaan internet paling banyak adalah untuk komunikasi melalui pesan diikuti dengan penggunaan media sosial. Berdasarkan survei yang dilakukan oleh *We Are Social*, bahwa per Januari 2019 pengguna aktif sosial media di Indonesia bertambah 15% atau sekitar 20 juta dari tahun sebelumnya sehingga pengguna aktif sosial media mencapai angka 150 juta. Twitter menempati peringkat enam teratas sebagai platform media sosial teraktif dengan angka mencapai 52% dan peringkat keempat sebagai media untuk berjejaring sosial atau *social networking*. Dari penelitian Kapoor et al. (2017), Twitter menjadi salah satu media berjejaring yang digunakan oleh

peneliti pada bidang penelitian media sosial. Banyak hal yang bisa diteliti dari kumpulan *tweet*. Salah satu yang populer dilakukan adalah menganalisa emosi dari *tweet*.

Berkembang pesatnya penggunaan media sosial membuat orang cenderung mengekspresikan emosi melalui media sosial. Emosi adalah perasaan yang kita alami. Kita menyebut berbagai emosi yang muncul dalam diri kita dengan berbagai nama seperti sedih, gembira, kecewa, semangat, marah, benci, cinta. Sebutan yang kita berikan kepada perasaan tertentu, mempengaruhi bagaimana kita berpikir mengenai perasaan itu, dan bagaimana kita bertindak. Emosi dasar manusia dapat dikategorikan menjadi marah, kesedihan, takut, kesenangan, cinta, kejutan, jijik, dan malu. Emosi dapat dideteksi melalui ekspresi suara, wajah, maupun tulisan. Pendeteksian emosi dapat dimanfaatkan untuk berbagai bidang.

Berdasarkan latar belakang yang telah dipaparkan, perlu dilakukan deteksi ekspresi emosi pada konteks tulisan atau teks dari sosial media Twitter guna memahami emosi yang tergambar melalui tulisan dalam tweet karena ekspresi wajah dan suara tidak dapat diketahui melalui tulisan. Merujuk pada penelitian yang dilakukan oleh [2], deteksi emosi pada tweet dilakukan dengan menggunakan *end-to-end deep learning* model yaitu *Sentiment and Semantic Based Emotion Detector* (SS-BED). SS-BED mengombinasikan algoritma *deep learning* yaitu LSTM dengan beberapa *word embedding* untuk mendeteksi emosi pada teks. Menggabungkan sentimen kata-kata yang berbeda dalam teks dan memahami semantik kalimat dapat meningkatkan akurasi klasifikasi emosi pada teks yang mengandung ambiguitas. Sebelumnya penulis akan menggunakan metode SS-BED, namun karena keterbatasan spek gadget, penelitian ini hanya menggunakan satu *layer*, yaitu GloVe.

Oleh karena itu, penelitian akan menggunakan metode LSTM-GloVe pada Implementasi *Deep Learning* Untuk Deteksi Ekspresi Emosi Pada Twitter dengan harapan deteksi emosi pada tweet dapat lebih akurat.

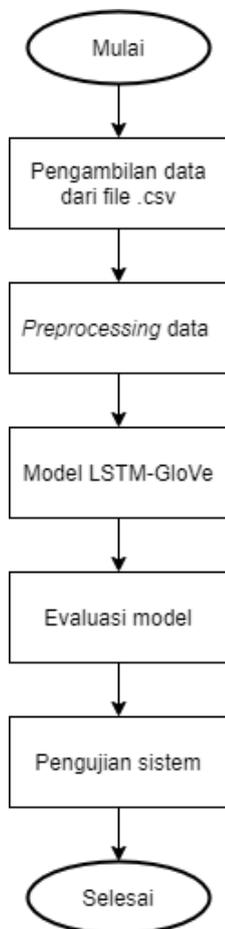
II. RELEVANSI PENELITIAN

Pada penelitian [3] mengklasifikasikan emosi dari Twitter berbahasa Indonesia hasilnya lebih akurat dengan menggunakan beberapa fitur yang sudah dikombinasikan dari fitur daftar kata-kata emosi, Bag-of-Words, dan FastText, Menambahkan sentimen dan *emoticon lexicon*, ortografi, serta fitur tag POS ke fitur kombinasi dasar dapat

meningkatkan kinerja dataset. Nilai F1 tertinggi dari kombinasi beberapa fitur ini adalah 69,73%.

Selanjutnya berdasarkan penelitian *Emotion Detection in Text using Nested Long Short-Term Memory* penelitian tersebut membandingkan metode Support Vector Machine (SVM), Long-Short Term Memory (LSTM), dan Nested LSTM. Metode Nested LSTM memiliki tingkat akurasi paling tinggi untuk mendeteksi multi-kelas emosi yaitu 99,167% dan untuk metode LSTM tidak berbeda jauh yaitu 99,154%. Namun LSTM memiliki performa yang lebih baik di *precision*, *recall*, dan *F1-score* [4]. Kemudian pada penelitian *Understanding Emotions in Text Using Deep Learning and Big Data*, deteksi emosi dilakukan dengan menggunakan model *Sentiment and Semantic Based Emotion Detector* (SS-BED). Nilai F1 yang didapat lebih besar dibanding dengan metode *machine learning* [2].

Dari penelitian-penelitian tersebut dapat dilihat bahwa penelitian yang menggunakan *machine learning* bisa menghasilkan nilai akurasi yang cukup baik. Namun tingkat akurasinya akan lebih baik bila menggunakan *deep learning*. Untuk itu dalam penelitian ini menggunakan model SS-BED yaitu mengkombinasikan metode *deep learning* dengan semantik dan sentimen agar mendapat akurasi dan hasil uji yang lebih baik lagi.

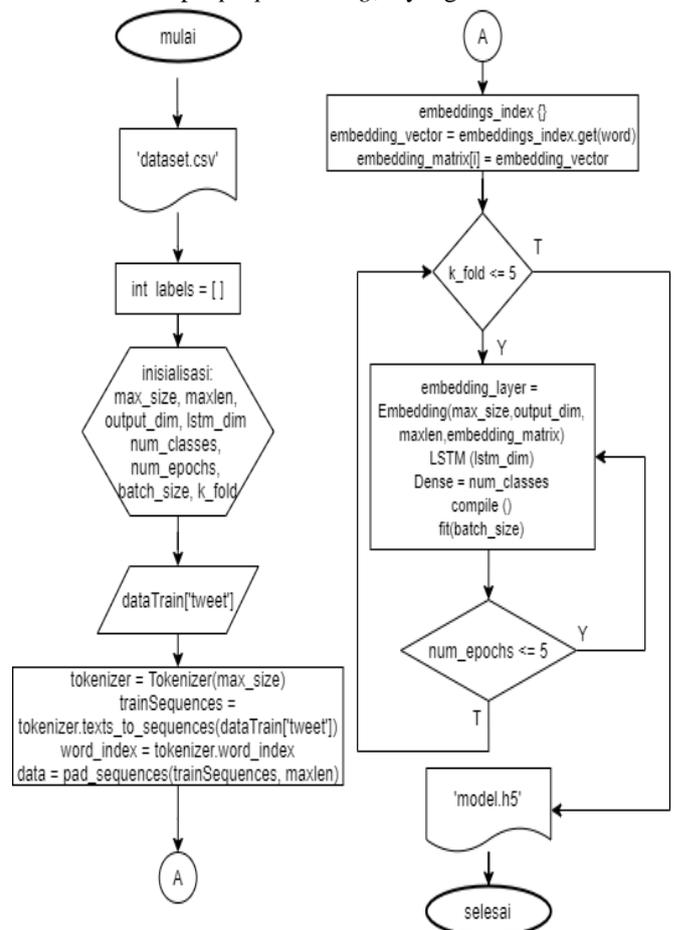


Gambar 1 Flowchart Umum Metode Pengolahan Data

III. METODOLOGI

Pada tahap desain sistem, proses pengolahan data secara keseluruhan dapat dilihat pada Gambar 1. Agar data *tweet*

yang akan digunakan dapat lebih mudah untuk diolah, data melewati tahap *preprocessing*, yang kemudian akan



digunakan pada model LSTM-GloVe. Model LSTM-GloVe

Gambar 2 Flowchart Model LSTM-GloVe yang telah dibuat kemudian akan dievaluasi menggunakan *k-fold cross-validation* untuk mendapatkan hasil akhir berupa *precision*, *recall*, dan *F-measure*.

Gambar 2 menunjukkan proses perancangan dan pembangunan model LSTM-GloVe dimana pertama sistem akan menginput data yang akan diolah dari dataset.csv yang berisi label dan *tweet*. Label dari dataset tersebut kemudian disimpan dalam sebuah array labels []. Kemudian sistem akan membuat variabel-variabel yang akan digunakan selama proses pengolahan data seperti *max_size*, *maxlen*, *output_dim*, *lstm_dim*, *num_classes*, *num_epochs*, *batch_size*, dan *k_fold*.

Data *tweet* yang sudah dilakukan pada tahap *preprocessing* kemudian akan diubah menjadi vektor. Sistem akan memproses class *Tokenizer* yang dapat membuat vektor dari *corpus* teks menjadi daftar bilangan *integer* dengan parameter berupa *max_size* (ukuran kamus). Kata-kata yang sudah dikonversi menjadi *sequence of integer* akan dijadikan sebagai kamus yaitu *word index*. Pengindeksan diurutkan sesuai dengan kata yang paling umum. Pada kamus *word index* kemudian akan diberi *padding*. Panjang kalimat yang kurang dari *maxlen* akan ditambahkan angka 0. Hal ini dilakukan agar panjang pada setiap data *tweet* menjadi sama

Sistem kemudian akan membangun indeks pada *embeddings_index* untuk mengambil indeks dari kamus

GloVe. Vektor tersebut kemudian oleh sistem disimpan dalam *embedding_vector*. Indeks pada *embedding_vector* dilakukan pemetaan dengan indeks pada *word_index* yang kemudian oleh sistem akan disimpan dalam *embedding_matrix*. Setelah dilakukan pemetaan indeks, kemudian dilakukan pembagian data antara data *training* dan data *testing* menggunakan *K-Fold Cross-validation* sebanyak 5 *k_fold*. Sistem akan melakukan pengecekan apakah nilai *k_fold* <= 5. Jika iya, sistem akan melanjutkan proses yaitu masuk ke *embedding layer* yang mengambil bilangan integer dari *embedding_matrix* dan memetakannya ke *dense vector* dari *embedding*. Pada *embedding layer* dibutuhkan parameter seperti *max_size* (ukuran kamus), *output_dim* (ukuran *dense vector*), dan *maxlen* (panjang *sequence*).

Output dari *embedding layer* berupa *dense vector* oleh sistem akan dimasukkan ke *layer LSTM* dengan ukuran units sebesar *lstm_dim*. *Output* dari layer tersebut akan diproses oleh sistem dengan masuk ke *layer* selanjutnya yaitu *dense layer*. Pada *layer* ini terdapat dua parameter yaitu parameter *num_classes* dan fungsi aktivasi yaitu *softmax*. Fungsi aktivasi akan menentukan hasil klasifikasi yaitu dengan menghitung probabilitas setiap kelas target dengan *range* 0 sampai 1. Langkah selanjutnya adalah sistem melakukan *compile* untuk mengubah urutan *sequential* yang menjadi serangkaian transformasi matriks yang efisien untuk dijalankan pada CPU. Proses setelah *compile* adalah *fit*, yaitu proses pembelajaran yang sebenarnya dengan menggunakan data latih. Data yang dilatih tidak dimuat semua sekaligus, melainkan dipartisi ke dalam kelompok pasangan pola input-output (*batch*) sesuai dengan *batch size* yang sudah ditentukan. Kemudian sistem akan melakukan pengecekan *epoch* mengenai pencapaian *epoch* ke-5. Jika *epoch* belum terpenuhi, maka sistem perlu mengulang ke proses *embedding layer*. Jika *epoch* terpenuhi yaitu mencapai *epoch* ke-5, maka sistem akan kembali ke pengecekan *k_fold*. Jika *k_fold* ke-5 sudah terpenuhi, maka sistem menyimpan model LSTM-GloVe yang sudah dilatih ke dalam *file* berbentuk *h5*.

IV. HASIL

Pengujian yang dilakukan adalah dengan parameter sebagai berikut:

1. *Learning rate*: 0,005 dan 0,001
2. *Dropout*: 0,25 dan 0,5
3. *Optimizer*: SGD dan ADAM

Untuk tiap parameter dilakukan sebanyak *k-fold* dengan nilai *k* sebesar 10, *epoch* sebesar 50 dan *batch size* sebesar 100. Pada penelitian ini pengujian dilakukan pada beberapa parameter untuk mengetahui nilai akurasi terbaik seperti berikut:

- a. Menganalisis parameter dengan jumlah *learning rate*
Berdasarkan Tabel 1, apabila data nomor 1 dibandingkan dengan data nomor 5 maka parameter dengan *learning rate* 0,005 memiliki tingkat akurasi lebih tinggi dan tingkat *loss* lebih rendah. Apabila data nomor 2 dibandingkan dengan data nomor 6 maka parameter dengan *learning rate* 0,01 memiliki tingkat akurasi lebih tinggi dan tingkat *loss* lebih rendah.
- b. Menganalisis parameter dengan jumlah *dropout*
Berdasarkan Tabel 1, apabila data nomor 1 dibandingkan dengan data nomor 3 maka parameter dengan *dropout* 0,25

memiliki tingkat akurasi lebih tinggi dan tingkat *loss* lebih rendah. Apabila data nomor 2 dibandingkan dengan data nomor 4 maka parameter dengan *dropout* 0,5 memiliki tingkat akurasi lebih tinggi dan tingkat *loss* lebih rendah.

- c. Menganalisis parameter dengan jenis *optimizer*
Berdasarkan tabel 1, *training* secara keseluruhan dengan menggunakan *optimizer* ADAM memiliki nilai akurasi lebih baik dibandingkan dengan *optimizer* SGD.

TABEL 1 KOMBINASI PARAMETER

No.	Lr. rate	Drop- out	Batch size	Ep- och	Opt.	Loss	Acc
1.	0,005	0,25	100	50	SGD	1,47	0,40
2.	0,005	0,25	100	50	ADAM	1,77	0,41
3.	0,005	0,5	100	50	SGD	1,48	0,36
4.	0,005	0,5	100	50	ADAM	1,29	0,50
5.	0,01	0,25	100	50	SGD	1,62	0,23
6.	0,01	0,25	100	50	ADAM	1,75	0,47
7.	0,01	0,5	100	50	SGD	1,47	0,38
8.	0,01	0,5	100	50	ADAM	1,28	0,48

Tidak ada aturan pasti dalam memilih nilai tiap parameter. Hasil yang terbaik dari deteksi emosi tidak tergantung dari banyaknya nilai tiap parameter. Tiap parameter memiliki ketergantungan satu sama lain. Pengujian pada penelitian ini mendapatkan hasil akurasi terbaik dengan menggunakan parameter learning rate sebesar 0,005, dropout sebesar 0,5, batch size sebesar 100, epoch sebesar 50 dan optimizer ADAM.

Pengujian performa pada sistem deteksi emosi dilakukan dengan menghitung nilai dari *precision*, *recall*, dan *F1 score* pada aplikasi. Didapatkan hasil rata-rata seperti pada Tabel 2:

TABEL 2 HASIL PENGUJIAN PERFORMA

Accuracy	Precision	Recall	F1-score
46%	33%	38%	35%

Berikut ini pada Tabel 3 merupakan salah satu contoh hasil deteksi emosi yang dihasilkan oleh model LSTM-GloVe.

TABEL 3 CONTOH HASIL DETEKSI EMOSI

Tweet	Label Deteksi	Label Asli
Soal jln Jatibaru,polisi tdk bs GERTAK gubernur .Emangny polisi tdk ikut pmbhasan? Jgn berpolitik. Pengaturan wilayah,hak gubernur. Persoalan Tn Abang soal turun temurun.Pelik.Perlu kesabaran.	Anger	Anger

Berdasarkan hasil akurasi pada setiap kelas, penelitian ini memiliki nilai lebih rendah pada *precision*, *recall*, dan *F-1 score* jika dibandingkan dengan penelitian (Chatterjee, et al., 2019) karena beberapa alasan sebagai berikut:

1. Data pada penelitian ini menggunakan data berupa tweet personal dalam Bahasa Indonesia dengan jumlah data 4.403 yang diambil pada tahun 2018 dengan 5 kelas emosi (*love*, *happy*, *anger*, *sadness*, dan *fear*).

Sedangkan dalam penelitian (Chatterjee, et al., 2019), data yang digunakan merupakan *tweet conversational pairs* (*tweet* beserta balasan *tweet*) dalam Bahasa Inggris dengan jumlah data 554.000 yang diambil dari tahun 2012-2015 dengan 3 kelas emosi (*happy, sad, angry*).

2. Data pada penelitian ini menggunakan Bahasa Indonesia dan kamus GloVe yang digunakan bersumber dari Wikipedia dalam Bahasa Inggris. Sedangkan dalam penelitian (Chatterjee, et al., 2019), data menggunakan Bahasa Inggris dan kamus GloVe yang bersumber dari Wikipedia dalam Bahasa Inggris.

V. KESIMPULAN

Metode LSTM-GloVe dapat digunakan untuk mendeteksi emosi pada Twitter dalam Bahasa Indonesia dengan melakukan *input* data berupa kalimat tweet dalam Bahasa Indonesia dengan melalui *preprocessing*, vektorisasi teks, dan LSTM-Glove yang menghasilkan *output* berupa nilai probabilitas dari setiap kelas. Nilai probabilitas tertinggi merupakan emosi yang tergambar pada tweet tersebut. Penggunaan metode LSTM-GloVe pada *tweet* berbahasa Indonesia memiliki tingkat akurasi paling tinggi mencapai 0,5 dengan menggunakan parameter *learning rate* sebesar 0,005, *dropout* sebesar 0,5, *batch size* sebesar 100, *epoch* sebesar 50 dan *optimizer* ADAM. Sedangkan tingkat *precision*, *recall*, dan *F1-score* mencapai 33%. Hal ini terjadi karena adanya *underfitting* selama pelatihan

DAFTAR PUSTAKA

- [1] A. M. Kaplan and M. Haenlein, "Users of the world, unite! the challenges and opportunities of social media," *Business Horizons*, vol. 53, pp. 59-68, 2010.
- [2] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley and P. Agrawal, "Understanding emotions in text using deep learning and big data," *Computers in Human Behavior*, 2019.
- [3] S. M. Saputri, R. Mahendra and M. Adrian, "Emotion classification on Indonesian twitter dataset," in *International Conference on Asian Language Processing (IALP)*, Bandung, 2018.
- [4] D. Haryadi and G. P. Kusuma, "Emotion detection in text using nested long short-term memory," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, 2019.