Stance Classification Pada Berita Berbahasa Indonesia Berbasis Bidirectional LSTM

Esther Irawati Setiawan, *Teknologi Informasi Institut Sains dan Teknologi Terpadu Surabaya*
Ika Lestari, *Magister Teknologi Informasi Institut Sains dan Teknologi Terpadu Surabaya*,

***Abstrak*—**Berita palsu masih menjadi masalah yang harus mendapat perhatian khusus. Media sosial, termasuk Facebook menjadi salah satu sarana yang mudah dan murah untuk menyebarkan suatu informasi yang bahkan belum tentu kebenarannya. Informasi tentang kesehatan menjadi salah satu topik berita palsu yang banyak tersebar ke masyarakat. Cara yang berbeda untuk mendeteksi berita palsu yaitu dengan menggunakan deteksi sikap (stance detection). Tujuan utama dari penelitian ini adalah merancang model yang memiliki kemampuan terbaik untuk melakukan tugas stance classification pada konteks bahasa Indonesia. Model ini diharapkan dapat digunakan untuk berkontribusi dalam menanggulangi masalah penyebaran berita palsu, khususnya di Indonesia. Metode BiLSTM dan GRU diusulkan untuk digunakan dalam melakukan stance classification terhadap headline berita dengan kelas for (mendukung), against (menentang), dan observing (netral). Stance classification pada penelitian ini menggunakan data sebanyak 3.941 headline berita yang terdiri dari 563 klaim dengan 7 tanggapan. Dataset dikumpulkan dari artikel-artikel berita kesehatan berbahasa Indonesia yang diposting pada laman Facebook. Model pada penelitian ini mampu menghasilkan akurasi F1-score paling tinggi sebesar 64% dengan FastText embedding. Metode GRU dapat menjadi salah satu pilihan tepat untuk melakukan stance classification dengan komputasinya yang lebih sederhana. Kinerja FastText jauh lebih unggul dibandingkan dengan Word2Vec dalam melakukan pembentukan vektor kata karena mampu mengatasi masalah out-of-vocabulary (OOV).

Kata Kunci—Berbahasa Indonesia, Berita Palsu***, Bidirectional LSTM, Stance classification, GRU***

I. PENDAHULUAN

Munculnya media baru seperti Twitter maupun Facebook, memungkinkan berita dapat dipublikasikan secara real-time[1]. Siapapun dapat menjadi pembuat berita dan memberikan dampak kepada masyarakat. Bahkan para pengumpul berita sering kali mengangkat cerita dari media sosial dan mempublikasikannya kembali tanpa pemeriksaan kebenarannya. Hal ini menyebabkan berita yang mengandung kebohongan (*hoax*) mudah sekali menyebar.

Teknologi seperti *Artificial Intelligent* (AI) dan *Natural Language Processing* (NLP) menawarkan solusi bagi para peneliti untuk membangun sistem yang secara otomatis dapat mendeteksi berita palsu.

Ika Lestari, Magister Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail : ika3@mhs.stts.edu)
Esther Irawati Setiawan, Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail : esther@stts.edu)

Namun, mendeteksi berita palsu adalah tugas yang menantang untuk diselesaikan, karena memerlukan model untuk meringkas berita dan membandingkannya dengan berita yang sebenarnya untuk dapat diklasifikasikan sebagai palsu. Selain itu, kesulitan membandingkan berita yang diusulkan dengan berita asli adalah karena sangat subjektif dan berpendirian[2]. Untuk itulah beberapa peneliti telah melakukan deteksi berita palsu dengan cara yang berbeda, yaitu menggunakan deteksi sikap (*stance*).

Salah satu topik berita palsu yang banyak tersebar ke masyarakat adalah informasi tentang kesehatan. Hal ini dikarenakan masyarakat kurang informasi terkait hal tersebut[3]. Para penyebar informasi merasa apa yang ia terima perlu untuk disebarkan kepada orang lain, tanpa mencari tahu terlebih dahulu kebenaran dari informasi tersebut. Media sosial menjadi sarana yang paling mudah dan murah untuk menyebarkan informasi. Bahkan platform media sosial menempati urutan tertinggi sebagai saluran penyebaran berita palsu[4]. Oleh karena itu, penelitian ini menggunakan data yang dikumpulkan dari media sosial yaitu Facebook untuk dilakukan proses klasifikasi sikap atau *stance classification*. *Stance classification* adalah proses klasifikasi terhadap sikap dari penulis teks dalam menanggapi suatu klaim berita atau informasi. Tanggapan yang diberikan dapat berupa mendukung (*for*), menentang (*against*), ataupun netral (*observing*).

Pada penelitian ini, metode *deep learning* yaitu *Bidirectional Long Short-Term Memory* (BiLSTM) dan *Gated Recurrent Unit* (GRU) diusulkan untuk digunakan dalam melakukan tugas klasifikasi sikap. Kedua model tersebut sangat ampuh dalam melakukan tugas pemrosesan data berurut. Banyak penelitian dalam bidang *stance classification* yang memanfaatkan model BiLSTM untuk melakukan proses klasifikasi, salah satunya adalah penelitian Damian [5]. Selain itu, penelitian lain yang memanfaatkan kedua model tersebut sekaligus adalah penelitian dari Gayathri [6]. Oleh karena itu, kedua model tersebut akan dimanfaatkan pada penelitian ini untuk melakukan klasifikasi sikap dalam konteks bahasa Indonesia.

Tujuan utama dari penelitian ini adalah merancang model yang memiliki kemampuan terbaik untuk melakukan tugas *stance classification* pada konteks bahasa Indonesia. Peneliti berharap model tersebut dapat digunakan untuk berkontribusi dalam menanggulangi masalah penyebaran berita palsu (*hoax*) di Indonesia.

II. PENELITIAN TERKAIT

Dengan kemajuan ilmu pengetahuan, banyak penelitian telah dilakukan terkait dengan *stance classification*. Tidak sedikit pula kompetisi diadakan guna menarik dan meningkatkan minat peneliti untuk ikut berkontribusi dalam menanggulangi masalah berita palsu. Beberapa penelitian yang telah dilakukan disajikan dalam bagian ini sebagai tinjauan literatur.

TABEL I
PENELITIAN TERKAIT STANCE CLASSIFICATION

No	Judul (tahun)	Metode	Hasil
1	*Stance Detection for the Fake News Challenge: Identifying Textual Relationships with Deep Neural Nets* (2017)	*LSTMs with Independent Encoding* *LSTMs with Conditional Encoding* *LSTMs with Bidirectional Conditional Encoding* *Sequence-to-sequence* *Recurrent Model with Attention*	Model ketiga memberikan kinerja terbaik, yaitu memperoleh akurasi klasifikasi lebih dari 97% pada dev set
2	*Stance Detection for Fake News Identification* (2017)	*conditioned bidirectional LSTM + global features*	Model mendapat skor keseluruhan sebesar 87.4% dan *mean F1 score* mencapai 69.5%
3	*Fake News Headline Classification using Neural Networks with Attention* (2017)	BiLSTM MLP dan BiLSTM Att MLP	Model BiLSTM + MLP mendapat akurasi tertinggi, yaitu sebesar 57% pada *batch size* 64, *embed dim* 300, dan *dense layer* 3
4	*Stance-In-Depth Deep Neural Approach to Stance Classification* (2018)	RNN, GRU, GRU + BiLSTM, BiGRU + BiLSTM, BiGRU, MLP, LSTM, dan BiLSTM	Model BiLSTM menghasilkan akurasi tertinggi yaitu sebesar 83.5%
5	Analisis Pendapat Masyarakat Terhadap Berita Kesehatan Indonesia Menggunakan Pemodelan Kalimat Berbasis LSTM (2020)	*Long Short-Term Memory (LSTM)*	Model LSTM mampu menghasilkan *mean F1 score* sebesar 71%
6	*Stance Classification* Post Kesehatan di Media Sosial dengan *FastText Embedding* dan *Deep Learning* (2020)	CNN, LSTM, BiLSTM dengan FastText dan Word2vec	Model dengan fastText mampu menghasilkan *F1 macro score* sebesar 64%

Penelitian yang dilakukan oleh Chaundry[7], Damian[5], Miller[8], dan Gayathri[6] menggunakan dataset dari *Fake News Challenge stage 1* (FNC-1) yaitu *headline* dan konten berita berbahasa Inggris. Dataset ini dapat diunduh pada halaman web *fakenewschallenge.org*. Keempat penelitian ini menggunakan *class output* sebanyak 4 (empat) kelas, yaitu *for, against, observing,* dan *unrelated.*

Pada penelitian Chaundry[7] dilakukan banyak percobaan dengan melakukan penyetelan beberapa hyperparameter. Hyperparameter utama yang digunakan dalam penelitian tersebut yaitu *learning rate*, panjang token dari *headline* dan artikel, penyertaan tanda baca, dan menjalankan klasifikasi dalam 2, 3, atau 4 kelas. Kemudian hasil dari penelitian ini yaitu model *LSTMs with Bidirectional Conditional Encoding* berhasil melakukan deteksi sikap (*stance*) dengan akurasi klasifikasi lebih dari 97% pada dev set. Sementara itu, pada penelitian Damian[5] model yang digunakan menghasilkan skor keseluruhan mencapai 87.4% dan *mean F1 score* mencapai 69.5%. Pada penelitian tersebut, *Glove* dengan dimensi 100 digunakan sebagai representasi vektor kata yang diperbarui selama *training* dan tidak menghapus *stopwords.* Kemudian panjang artikel untuk input ke model adalah maksimal 200 kata, sedangkan panjang *headline* tidak dilakukan pemotongan.

Pada penelitian Miller[8], *Glove* juga digunakan sebagai representasi vektor kata dengan jumlah dimensi 100 dan 300. Beberapa penyetelan hyperparameter juga dilakukan pada penelitian ini hingga model BiLSTM + MLP yang digunakan berhasil menghasilkan akurasi tertinggi, yaitu sebesar 57% pada *batch size* 64, *embed dim* 300, dan *dense layer* 3. Selanjutnya pada penelitian Gayathri[6], keseluruhan model yang disebutkan pada tabel 1 dibandingkan, kemudian dianalisa dampak dari jumlah *layer* dan dilakukan pengukuran waktu *training* yang diperlukan oleh setiap model. Hasil dari penelitian ini menunjukkan bahwa model BiLSTM mampu menghasilkan akurasi tertinggi yaitu sebesar 83.5%.

Berbeda dengan keempat penelitian sebelumnya, pada penelitian yang dilakukan oleh Setiawan[9] dan Lim[10], dataset yang digunakan adalah berita berbahasa Indonesia. Selain perbedaan pada dataset, *class output* yang digunakan juga berbeda. Jika pada keempat penelitian sebelumnya menggunakan 4 *class output*, pada penelitian Setiawan[9] dan Lim[10] *class output* yang digunakan sebanyak 3 (tiga) kelas, yaitu *for, against,* dan *observing.* Pada penelitian Setiawan [9], model LSTM yang digunakan berhasil melakukan deteksi sikap (*stance*) dengan *mean F1 score* sebesar 71%. Sementara itu, model yang digunakan pada penelitian Lim[10] yaitu BiLSTM dengan *word embedding* FastText menghasilkan akurasi *F1-score* sebesar 64%.

Penelitian ini merupakan pengembangan penelitian dalam bidang *stance classification* berbahasa Indonesia. Pengembangan yang dilakukan adalah dengan menggunakan model GRU sebagai *classifier* yang kemudian dibandingkan kinerjanya dengan model BiLSTM. Hal ini dilakukan untuk mengetahui model mana yang lebih akurat dalam menentukan *stance* dari sebuah judul berita yang menanggapi klaim. Selain itu, *classifier stance* dengan menggunakan model GRU belum pernah diuji coba pada penelitian Setiawan[9] dan Lim[10].

A. *Stance Classification*

Stance classification merupakan sub-domain dari sentiment analysis. *Stance classification* didefinisikan

sebagai tugas untuk mengklasifikasi hubungan antara dua teks. Dalam penelitian ini, hubungan antara dua teks tersebut dikelompokkan menjadi 3 (tiga), yaitu mendukung (*for*), menentang (*against*), dan netral (*observing*). Penelitian – penelitian *stance classification* sebelumnya digunakan untuk mendeteksi sikap peserta dalam debat online[11], esai mahasiswa, dan debat kongres[12]. Penelitian tersebut bekerja dengan adanya target yang spesifik. Berbeda dengan penelitian yang dilakukan pada sebuah kompetisi bernama *fake news challenge stage 1* (FNC-1) dengan topik *stance detection.* Pada kompetisi ini, dataset yang digunakan adalah *emergent dataset*[1]. Dataset ini terdiri dari sekitar 50.000 pasangan *headline*-artikel yang masing – masing diberi label dengan *unrelated, discuss, agree,* atau *disagree.*

B. Bidirectional Long Short-Term Memory (BiLSTM)

Pada bidang *Natural Language Processing* (NLP), khususnya *stance classification* atau *stance detection* telah banyak yang menggunakan model biLSTM. Yuanyu[13] telah berhasil menggunakan model biLSTM untuk melakukan *stance detection* pada dataset Semeval-2016 Task 6.A (*English dataset*) dan NLPCC-2016 *Stance Detection Shared Task* (*Chinese dataset*). Selain itu, Isabelle[14] juga menggunakan model biLSTM pada dataset SemEval 2016 Task 6 *Twitter Stance Detection corpus*. Kemudian pada dataset yang berbeda yaitu emergent[1] FNC-1, Damian[5] dan Miller[8] juga telah berhasil menggunakan model biLSTM dalam penelitiannya.

C. Gated Recurrent Unit (GRU)

Selain menggunakan model BiLSTM, model lain yang sering digunakan dalam penelitian bidang NLP adalah GRU. Pada penelitian identifikasi berita palsu, model GRU mampu menghasilkan akurasi yang tidak kalah baik dengan BiLSTM. Gayathri[6] telah berhasil menerapkan model GRU untuk melakukan *stance classification* dengan emergent dataset (FNC-1)[1]. Selain itu, Reddy[15] juga telah memanfaatkan model GRU untuk mengikuti kompetisi UrduFake FIRE-2020 dengan menggunakan dataset Urdu Fake News Dataset.

III. GAMBARAN DATASET

Dataset yang digunakan dalam penelitian ini adalah *Indonesian News Stance Dataset*[9] yang diperbarui pada penelitian Lim[10]. Pembaruan dataset ini dilakukan dengan mengumpulkan artikel – artikel berita kesehatan berbahasa Indonesia yang dipost pada laman Facebook. Pengambilan dataset tersebut adalah dari judul artikel yang dibagikan, judul pada gambar yang dipost, atau teks judul pada post. Dataset terdiri dari id, judul berita (*headline*) sebagai klaim, sumber, jumlah *like*, jumlah *comment*, judul berita (*headline*) lain sebagai tanggapan, id tanggapan, sumber tanggapan, jumlah *like* tanggapan, jumlah *comment* tanggapan, dan kelas *stance*. Kemudian dataset ini diberi label secara manual. Label *stance* yang digunakan meliputi :

- *For* : *headline* tanggapan mendukung *headline* klaim
- *Against* : *headline* tanggapan menentang *headline* klaim
- *Observing* : *headline* tanggapan tidak memberikan penilaian terhadap *headline* klaim atau bersifat netral.

Dalam melakukan uji coba pada penelitian ini, dilakukan penambahan satu judul artikel yang menanggapi setiap klaim. Penambahan ini dilakukan untuk mengetahui dampak dari jumlah data terhadap akurasi yang dihasilkan oleh model. Akibatnya jumlah dataset yang digunakan dalam penelitian ini adalah sebanyak 3.941 *headline* berita. Dataset terdiri dari 563 *headline* klaim, dan 7 *headline* berita lain yang menanggapi. Pada awalnya, jumlah dataset yang digunakan pada penelitian Lim[10] adalah sebanyak 3.378 *headline* berita, dengan 6 *headline* berita lain yang menanggapi klaim. Distribusi dataset ini ditunjukkan pada tabel II. Contoh dataset yang digunakan dalam penelitian ini ditunjukkan pada tabel III. Pada penelitian ini, data yang digunakan hanya id, *headline* klaim, *headline* tanggapan, dan kelas *stance.*

TABEL II
DISTRIBUSI DATASET

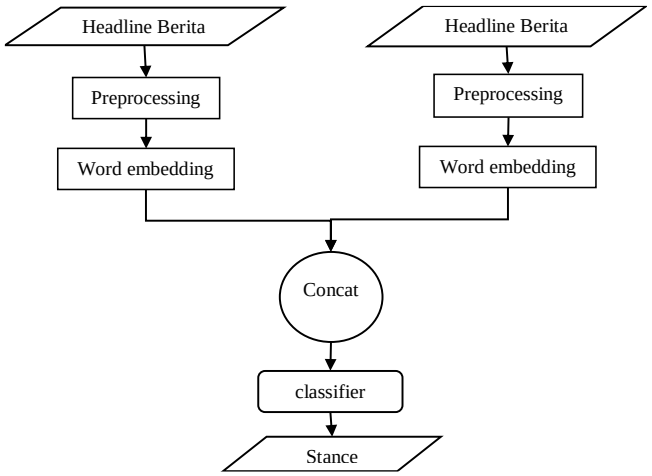
No	Stance	Jumlah	Persentase (%)
1	*Against*	1.275	32.4
2	*For*	1.391	35.3
3	*Observing*	1.275	32.4

TABEL III
CONTOH DATASET

Headline Klaim	Headline Tanggapan	Stance
Awas, Ini Bahaya Tidur Siang yang Lebih dari Sejam	Waspada Risiko Penyakit Ini Jika Tidur Siang Terlalu Lama	for
	Terlalu Lama Tidur Siang Memicu Penyakit Jantung	for
	Penting gak sih tidur siang bagi si kecil???	observing
	Kapan Waktu yang Tepat untuk Tidur Siang?	observing
	7 Manfaat Tidur Siang yang Tak Banyak Diketahui	against
	MANFAAT TIDUR SIANG	against
	10 Manfaat Tidur Siang, Sangat Baik untuk Menjaga Kondisi Tubuh	against

IV. METODOLOGI PENELITIAN

Bidirectional long short-term memory (*Bi-LSTM*) telah sukses digunakan untuk tugas pemrosesan bahasa alami. Dengan dataset emergent, Damian[5] telah berhasil melakukan deteksi sikap dengan menggunakan model Bi-LSTM dan menghasilkan *score* sebesar 87.4% dan *mean F1 score* sebesar 69.5%. Berdasarkan hal ini, penulis mengadopsi model Bi-LSTM untuk melakukan deteksi sikap (*stance detection*) pada berita berbahasa Indonesia. Pada penelitian ini diberikan batasan yaitu dengan tidak melakukan proses klasifikasi untuk data dengan kedua input *headline* berita membahas topik yang berbeda (*unrelated*). Alur sistem dari penelitian ini ditunjukkan pada gambar 2.



Gambar. 2. Alur Sistem Penelitian

A. *Preprocessing*

Setiap input yaitu *headline* klaim dan *headline* tanggapan akan dilakukan preprocessing terlebih dahulu. Tahapan preprocessing yang dilakukan dalam penelitian ini terdiri dari *case folding*, *tokenizing*, *filtering*, dan *stemming*. *Case folding* yaitu proses penyamaan case dalam sebuah dokumen (*lowercase*). *Tokenizing* yaitu proses memisahkan deretan kata dalam kalimat menjadi token atau potongan kata tunggal. Proses *tokenizing* dilakukan dengan menggunakan *word_tokenize* yang dimiliki oleh *library* nltk. *Filtering* yaitu proses membuang kata – kata tidak penting dari hasil token, seperti "yang", "di", "dan", dan lain - lain. Proses *filtering* ini berdasarkan *stopwords* dari Tala[16]. Selain menghapus kata yang tidak penting, dalam proses *filtering* juga dilakukan penghapusan karakter spesial kecuali tanda tanya, tanda hubung, dan angka 0 – 9. Dan yang terakhir yaitu *stemming*, adalah proses pengembalian suatu kata berimbuhan ke bentuk dasarnya. Proses *filtering* dan *stemming* yang dilakukan dalam penelitian ini menggunakan library Sastrawi.

B. *Word Embedding*

Setelah input melalui tahap preprocessing, selanjutnya dilakukan *word embedding*. Dalam penelitian ini, *word embedding* yang digunakan adalah Word2Vec dan FastText. Algoritma Word2Vec diciptakan oleh Mikolov[17] pada tahun 2013. Dalam penelitian ini, Word2Vec yang digunakan adalah *pre-trained* Word2Vec dengan ukuran dimensi 300. *Pre-trained* Word2Vec ini dapat diakses pada[18]. Sedangkan FastText *embedding* yang digunakan adalah cc.id.300.bin dengan ukuran dimensi 300. FastText *embedding* merupakan pengembangan dari Word2Vec. Algoritma ini diciptakan oleh Bojanowski[19] pada tahun 2017.

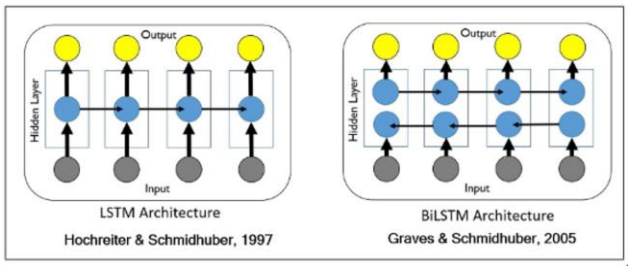
Pada tahap ini, sebuah input berupa teks atau string akan dikonversi menjadi angka atau vektor. Hal ini agar input tersebut dapat diolah oleh arsitektur *deep learning*. Selain input harus berupa angka atau vektor, input harus memiliki bentuk (*shape*) dan ukuran (*size*) yang sama. Namun secara alamiah, setiap kalimat input tidak memiliki jumlah kata yang sama. Oleh karena itu, diperlukan proses *padding* menggunakan *pad_sequences* milik Tensorflow. Kita juga dapat menentukan jumlah maksimal kata untuk setiap kalimat. Sebelum dilakukan proses *padding*, kalimat input perlu ditokenisasi terlebih dahulu menggunakan *tokenizer tool* milik Tensorflow.

C. *Classifier*

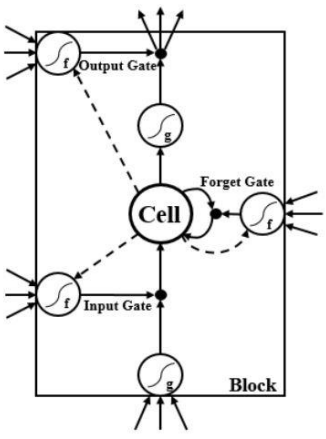
1) *Klasifikasi dengan Bidirectional LSTM (BiLSTM)*

Setelah tahap *word embedding* selesai dilakukan, kemudian kedua vektor dari *headline* berita akan digabung (*concat*) untuk dijadikan sebagai input dari *classifier*. Proses klasifikasi dengan BiLSTM adalah untuk mentukan sikap (*stance*) dari *headline* berita tanggapan terhadap *headline* berita klaim. Sikap (*stance*) yang digunakan dalam penelitian ini adalah *for*, *against*, dan *observing*.

Arsitektur dari BiLSTM terdiri dari *forward* LSTM dan *backward* LSTM. BiLSTM dapat menyesuaikan data dari arah maju (*forward*) dan mundur (*backward*), kemudian menggabungkan prediksi. *Forward* dan *backward* pada BiLSTM dapat meningkatkan jumlah informasi yang tersedia ke jaringan dan konteks yang tersedia untuk algoritma, misalnya mengetahui kata apa yang segera mengikuti dan mendahului kata dalam kalimat. Untuk memahami kata dalam NLP, terkadang tidak hanya dibutuhkan kata sebelumnya, melainkan juga kata yang akan datang. Perbedaan arsitektur antara LSTM dan BiLSTM ditunjukkan pada gambar 3[20]. Sementara itu, struktur dari satu sel LSTM ditunjukkan pada gambar 4[21].



Gambar. 3. Perbedaan Arsitektur antara LSTM dan BiLSTM



Gambar. 4. Struktur satu sel LSTM

LSTM memiliki *cell memory* yaitu tambahan informasi sinyal yang diberikan dari satu *time step* ke *time step* berikutnya. Satu sel LSTM memiliki 3 (tiga) mekanisme gerbang, yaitu *forget gate*, *input gate*, dan *output gate*. Langkah kerja dari LSTM adalah dimulai melalui komponen

forget gate (f_t) yang ditunjukkan pada (1). Pada gerbang ini ditentukan informasi yang tidak dibutuhkan agar dibuang dari *memory cell* (C_{t-1}) dengan menggunakan fungsi sigmoid. Pada *forget gate*, nilai vektor *hidden state* dalam *time step* sebelumnya (h_{t-1}) dan nilai vektor input x dalam *time step t* (x_t) dibaca, kemudian menghasilkan angka antara 0 (artinya lupakan elemen) dan 1 (artinya jaga elemen) untuk setiap elemen dalam C_{t-1}.

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

Langkah selanjutnya adalah menentukan informasi yang akan diperbarui melalui *input gate* (i_t). Pada saat yang sama, juga dihasilkan kandidat vektor *memory cell* yang baru ($\tilde{C}_t$). Hal ini ditunjukkan pada (2) dan (3). Hasil dari kedua proses ini kemudian digabungkan untuk mengupdate C_t. Dalam melakukan update C_t dari C_{t-1} dilakukan perkalian antara C_{t-1} dengan f_t untuk melupakan informasi, kemudian mengalikan $\tilde{C}_t$ dengan i_t untuk memutuskan seberapa banyak kandidat C_t disertakan, lalu keduanya dijumlahkan. Proses update ini ditunjukkan pada (4).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Kemudian langkah terakhir dari LSTM adalah menentukan *output* yang didasarkan pada nilai C dan dilewatkan ke suatu filter. Dalam langkah ini terdapat 3 tahapan. Pertama, menjalankan gerbang sigmoid yang disebut *output gate* (o_t) untuk memutuskan bagian – bagian apa dari C yang akan dihasilkan (5). Kedua, melewatkan C melalui tanh untuk membuat nilainya menjadi antara -1 dan 1. Terakhir, mengalikan *output* gerbang sigmoid tadi sehingga menghasilkan bagian yang diputuskan sesuai dengan (6).

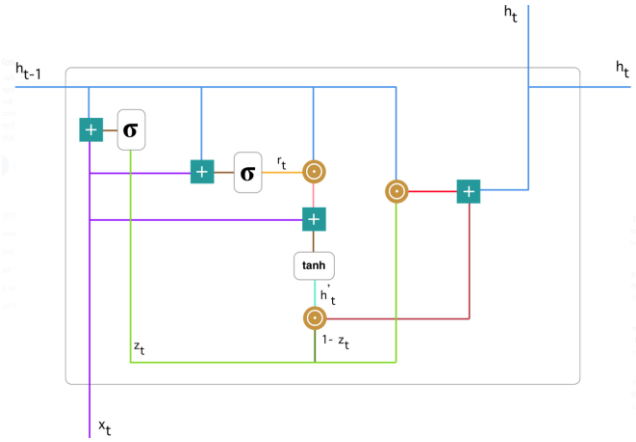
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$s_t = o_t * \tanh(C_t) \quad (6)$$

2) Klasifikasi dengan Gated Recurrent Unit (GRU)

GRU juga digunakan sebagai *classifier* dalam penelitian ini. Input yang digunakan pada model GRU adalah teks yang telah melalui proses *preprocessing* dan *word embedding*, yakni berupa vektor dari *headline* berita. Model GRU akan melakukan proses klasifikasi, yaitu apakah *headline* berita tanggapan mengandung salah satu sikap (*stance*) terhadap *headline* berita klaim.

Model GRU pertama kali diperkenalkan oleh Cho[22] dan Chung[23] pada tahun 2014. Model ini diciptakan untuk mengatasi masalah *vanishing gradient* pada *Recurrent Neural Network* (RNN). Model GRU dipilih untuk digunakan dalam penelitian ini adalah karena GRU memiliki rancangan yang hampir sama dengan LSTM, bahkan terkadang hasil dari kedua model ini juga sama baiknya. Arsitektur dari GRU ditunjukkan pada gambar 5[24].



Gambar. 5. Struktur satu sel GRU

Dalam satu struktur sel GRU terdapat 2 *gate*, yaitu *update gate* dan *reset gate* yang digunakan untuk mengatasi masalah *vanishing gradient* dari struktur RNN biasa. Langkah kerja dari GRU adalah dimulai dengan menghitung *update gate* (z_t) untuk langkah waktu t dengan formula ditunjukkan pada (7). Pada langkah ini, x_t yang masuk ke jaringan akan dikalikan dengan bobotnya sendiri ($W_{(z)}$). Hal ini juga berlaku untuk $h_{(t-1)}$ yang menyimpan informasi untuk unit $t-1$ sebelumnya dan dikalikan dengan bobotnya sendiri ($U_{(z)}$). Hasil dari kedua proses ini kemudian dijumlahkan dan fungsi aktivasi *sigmoid* diterapkan. Hasil dari *update gate* adalah antara 0 dan 1. Fungsi dari *update gate* ini adalah untuk membantu model menentukan berapa banyak informasi masa lalu akan diteruskan ke masa depan. Sehingga model dapat menyalin semua informasi masa lalu dan menghilangkan masalah *vanishing gradient.*

$$z_t = \sigma(W^{(z)} x_t + U^{(z)} h_{t-1}) \quad (7)$$

Langkah selanjutnya adalah *reset gate* yang berfungsi untuk membantu model melupakan informasi masa lalu. Formula dari *reset gate* ditunjukkan pada (8). Formula ini hampir sama dengan *update gate*, hanya berbeda pada bobotnya. Kemudian pada konten memori baru akan menggunakan *reset gate* untuk menyimpan informasi yang relevan dari masa lalu dengan formula (9). Langkah terakhir, jaringan perlu menghitung h_t (vektor yang menyimpan informasi untuk unit saat ini) dan meneruskannya ke jaringan. Untuk melakukan hal tersebut, langkah ini memerlukan *update gate*. Fungsi dari langkah terakhir ini adalah untuk menentukan informasi yang harus dikumpulkan dari konten memori saat ini (h'_t) dan dari langkah sebelumnya (h_{t-1}). Formula dari langkah ini ditunjukkan pada (10).

$$r_t = \sigma(W^{(r)} x_t + U^{(r)} h_{t-1}) \quad (8)$$

$$h'_t = \tanh(W_{x_t} + r_t \circ U h_{t-1}) \quad (9)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ h'_t \quad (10)$$

D. Sistem Evaluasi Performa

Dalam penelitian ini, akurasi diukur menggunakan *F1-score* dari setiap label dan *macro-average F1-score*. *F1-score* diperoleh dari *harmonic mean* antara *precision* dan *recall*. Rentang nilai dari *F1-score* adalah antara 0 hingga 1. Perhitungan *F1-score* dapat dilihat pada (11).

$$F1-score=\frac{2*precision*recall}{precision+recall} \quad (11)$$

Precision adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Sementara, *recall* adalah persentase item positif diidentifikasi dengan benar. Perhitungan *precision* dan *recall* dapat dilihat pada (12) dan (13).

$$precision=\frac{tp}{tp+fp} \quad (12)$$

$$recall=\frac{tp}{tp+fn} \quad (13)$$

dimana tp (*true positive*) adalah banyaknya tebakan positif dengan *target class* sebenarnya positif, fp (*false positive*) adalah banyaknya tebakan positif padahal *target class* yang sesungguhnya adalah negatif, sedangkan fn (*false negative*) adalah banyaknya tebakan negatif dengan *actual class* positif.

V. Hasil Eksperimen

Dalam penelitian *stance classification* ini dilakukan percobaan dengan beberapa parameter dan metode yang berbeda. Dataset yang digunakan untuk melakukan uji coba adalah 20% dari keseluruhan data. Dalam penelitian ini, *macro-average F1-score* digunakan untuk mengetahui akurasi prediksi dari semua label. Sementara itu, *F1-score* digunakan untuk mengetahui akurasi prediksi pada setiap label. Proses percobaan pada penelitian ini dilakukan dengan menguji coba model sebanyak sepuluh *seed*, sehingga terdapat 10 *classification report*. Kemudian hasil uji coba yang digunakan adalah rata – rata *F1-score* dan *macro-average F1-score* dari 10 *classification report* tersebut. Hal tersebut juga dilakukan pada penelitian Lim[10]. Parameter yang digunakan untuk melakukan uji coba terdiri dari: jumlah dataset, jenis *word embedding*, jenis *classifier*, dan *dropout rate*. Beberapa parameter yang diuji coba dalam penelitian ini mengacu dari penelitian Mrowca[5] dan Lim[10].

A. Jumlah Dataset

Seperti yang telah dibahas pada section III tentang gambaran dataset bahwa dilakukan penambahan *headline* tanggapan untuk penelitian ini. Pada penelitian ini dilakukan perbandingan hasil akurasi model dengan menggunakan data sebanyak 3.378 (data asli dari penelitian Lim[10]) dan 3.941 (dataset sudah ditambah). Hasil penelitian ditunjukkan pada tabel IV. Hasil penelitian menunjukkan bahwa dengan penambahan dataset sebanyak 563 data tidak berpengaruh secara signifikan terhadap hasil akurasi model. Dengan melakukan penambahan dataset hanya mampu meningkatkan akurasi *F1-score* sebanyak 2% dari 62% menjadi 64% pada model BiLSTM dan GRU.

B. Jenis Word Embedding

Jenis *word embedding* yang digunakan dalam penelitian ini adalah *pre-trained* Word2Vec dan FastText dengan dimensi keduanya adalah 300. Penggunaan Word2Vec sebagai *word embedding* dalam penelitian ini adalah mengacu pada penelitian Gayathri[6] dan Setiawan[9]. Sedangkan FastText mengacu pada penelitian Lim[10] yang mana cc.id.300.bin cocok digunakan untuk *stance classification* dengan fitur kalimat.

Hasil uji coba untuk perbandingan kinerja *word embedding* yang digunakan dalam penelitian ini ditunjukkan pada tabel V. Pada tabel tersebut terlihat bahwa *word embedding* dengan menggunakan FastText mampu menghasilkan akurasi *F1-score* sebesar 64%, jauh lebih tinggi dibandingkan dengan Word2Vec yang hanya mampu menghasilkan akurasi *F1-score* sebesar 51%. Kemudian dengan menggunakan Word2Vec, GRU mampu lebih unggul 6% dibandingkan dengan BiLSTM.

C. Jenis Classifier

Selain melakukan percobaan dengan *word embedding* yang berbeda, penelitian ini juga melakukan percobaan dengan membandingkan dua jenis *classifier* yang berbeda. Jenis *classifier* tersebut yaitu BiLSTM dan GRU. Pada penelitian Gayathri[6], model BiLSTM mampu menghasilkan akurasi paling besar dibandingkan dengan GRU. Namun pada penelitian ini, tidak ada perbedaan hasil akurasi dari model BiLSTM dan GRU. Kedua model tersebut mampu menghasilkan akurasi *F1-score* sebesar 64%. Hasil penelitian ini dapat dilihat pada tabel VI.

D. Dropout Rate

Penelitian ini menguji coba *dropout rate* dengan nilai 0.25, 0.5, 0.75, dan 0.99. Keempat nilai tersebut diuji cobakan kepada BiLSTM dan GRU dengan *FastText embedding*. Hasil penelitian ini ditunjukkan pada tabel VII. Dari tabel tersebut terlihat bahwa pada model BiLSTM dn GRU, nilai *dropout rate* yang semakin tinggi justru memperburuk akurasi model yang dihasilkan.

Pada *dropout rate* dengan nilai 0.75, akurasi *F1-score* pada model GRU mengalami penurunan sebesar 1% dari 64% menjadi 63%. Kemudian setelah nilai *dropout rate* dinaikkan menjadi 0.99, justru akurasi *F1-score* model GRU mengalami penurunan drastis sebesar 4% dari 63% menjadi 59%. Hal ini juga berlaku pada model BiLSTM, nilai *dropout rate* sebesar 0.99 juga memperburuk akurasi *F1-score*. Pada *dropout rate* 0.99 akurasi model BiLSTM mengalami penurunan sebesar 2% dari 64% menjadi 62%.

TABEL IV
HASIL UJI COBA DENGAN JUMLAH DATASET YANG BERBEDA

Jumlah Dataset	*F1 for*	*F1 against*	*F1 observing*	*F1-score*	*Macro-avg F1-score*
3.378					
BiLSTM	**0.58**	0.50	0.77	**0.62**	**0.62**
GRU	0.52	**0.51**	**0.81**	**0.62**	0.61
3.941					
BiLSTM	0.60	0.50	**0.81**	**0.64**	**0.64**
GRU	**0.61**	**0.53**	0.79	**0.64**	**0.64**

TABEL V
HASIL UJI COBA DENGAN JENIS WORD EMBEDDING

Word Embedding	*F1 for*	*F1 against*	*F1 observing*	*F1-score*	*Macro-avg F1-score*
Word2Vec + BiLSTM	0.45	0.41	0.55	0.48	0.47
Word2Vec + GRU	0.53	0.51	0.56	0.54	0.53
Rata - rata				**0.51**	**0.50**
FastText + BiLSTM	0.60	0.50	0.81	0.64	0.64
FastText + GRU	0.61	0.53	0.79	0.64	0.64
Rata - rata				**0.64**	**0.64**

TABEL VI
HASIL UJI COBA DENGAN JENIS CLASSIFIER YANG BERBEDA

Jenis Classifier	*F1 for*	*F1 against*	*F1 observing*	*F1-score*	*Macro-avg F1-score*
BiLSTM	0.60	0.50	**0.81**	**0.64**	**0.64**
GRU	**0.61**	**0.53**	0.79	**0.64**	**0.64**

TABEL VII
HASIL UJI COBA DENGAN PARAMETER DROPOUT RATE

Dropout Rate	*F1 for*	*F1 against*	*F1 observing*	*F1-score*	*Macro-avg F1-score*
BiLSTM					
0.25	0.60	0.51	0.80	**0.64**	**0.64**
0.5	0.60	0.50	**0.81**	**0.64**	**0.64**
0.75	0.59	**0.53**	0.80	**0.64**	**0.64**
0.99	**0.61**	0.41	0.79	0.62	0.60
GRU					
0.25	0.60	0.51	**0.81**	**0.64**	**0.64**
0.5	**0.61**	**0.53**	0.79	**0.64**	**0.64**
0.75	0.60	0.49	0.80	0.63	0.63
0.99	0.60	0.37	0.77	0.59	0.58

Hasil akhir dari penelitian ini memperlihatkan bahwa GRU mampu memberikan hasil yang setara dengan BiLSTM. Jika dibandingkan dengan penelitian sebelumnya, yaitu penelitian Lim[10], model BiLSTM memberikan hasil yang sama yaitu akurasi *F1-score* sebesar 64%. Namun jika dibandingkan dengan penelitian yang dilakukan oleh Setiawan[9], hasil penelitian ini menunjukkan adanya penurunan drastis pada model dengan *word embedding* Word2Vec. Hasil akhir dari penelitian Setiawan[9] adalah akurasi *F1-score* sebesar 71%, sedangkan pada penelitian ini, model dengan *word embedding* Word2Vec mampu menghasilkan akurasi *F1-score* sebesar 51%. Penurunan sebanyak 20% ini disebabkan karena Word2Vec yang digunakan pada penelitian Setiawan[9] adalah *on-trained* Word2Vec yang mana representasi vektor yang dihasilkan lebih disesuaikan terhadap arsitektur tugas *stance classification* yang digunakan.

VI. KESIMPULAN DAN SARAN

Stance classification pada berita berbahasa Indonesia menghasilkan akurasi tertinggi sebesar 64% dengan menggunakan algoritma *bidirectional LSTM* (Bi-LSTM) maupun *Gated Recurrent Unit* (GRU). Kedua model ini menghasilkan nilai akurasi *F1-score* yang sama dengan menggunakan *word embedding* FastText. Dengan demikian, model GRU yang mana merupakan varian RNN dengan komputasi sederhana dapat menjadi salah satu pilihan tepat untuk melakukan tugas klasifikasi, khususnya *stance classification.* Kemudian performa kinerja dari FastText jauh lebih akurat dibandingkan dengan Word2Vec. Hal ini disebabkan karena FastText *embedding* memiliki kemampuan dalam menangani kata – kata yang belum pernah ditemui sebelumnya atau yang disebut dengan *out-of-vocabulary* (OOV).

Penelitian mengenai *stance classification* pada dataset bahasa Indonesia ini memiliki peluang besar untuk terus dikembangkan. Salah satu pengembangan yang dapat dilakukan adalah dengan mencoba metode representase kata yang lain seperti Bert *Embedding*.

DAFTAR PUSTAKA

[1] W. Ferreira and A. Vlachos, “Emergent: a novel data-set for stance classification,” in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2016, pp. 1163–1168.

[2] A. Thota, “Fake News Detection : A Deep Learning Approach,” *SMU Data Sci. Rev. Vol. 1 No. 3, Artic. 10*, vol. 1, no. 3, 2018.

[3] A. M. Hasan, “Info Hoax Soal Kesehatan Paling Banyak Beredar di Masyarakat,” *https://tirto.id/*, 2017.

[4] “Hasil Survey Wabah HOAX Nasional 2019,” *https://mastel.id/*, 2019.

[5] D. Mrowca and E. Wang, “Stance detection for fake news identification,” *Eliaswang.Com*, 2017.

[6] G. Rajendran, B. Chitturi, and P. Poornachandran, “Stance-In-Depth Deep Neural Approach to Stance Classification,” *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1646–1653, 2018, doi: 10.1016/j.procs.2018.05.132.

[7] P. Chaudhry, A. K., Baker, D. & Thun-Hohenstein, “Stance Detection for the Fake News Challenge: Identifying Textual Relationships with Deep Neural Nets,” *Stanford*, pp. 1–10, 2017.

[8] K. Miller and A. Oswalt, “Fake News Headline Classification using Neural Networks with Attention.”

[9] E. I. Setiawan *et al.*, “Analisis Pendapat Masyarakat terhadap Berita Kesehatan Indonesia menggunakan Pemodelan Kalimat berbasis LSTM (Indonesian Stance Analysis of Healthcare News using Sentence Embedding Based on LSTM),” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, no. 1, pp. 8–17, 2020.

[10] E. Lim, E. I. Setiawan, and J. Santoso, “Stance Classification Post Kesehatan di Media Sosial Dengan FastText Embedding dan Deep Learning,” pp. 65–73, 2020.

[11] K. S. Hasan and V. Ng, “Stance classification of ideological debates: Data, models, features, and constraints,” in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 1348–1356.

[12] M. Thomas, B. Pang, and L. Lee, “Get out the vote: Determining support or opposition from Congressional floor-debate transcripts,” *COLING/ACL 2006 - EMNLP 2006 2006 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, no. July, pp. 327–335, 2006.

[13] Y. Yang, B. Wu, K. Zhao, and W. Guo, “Tweet stance detection: A two-stage DC-BILSTM model based on semantic attention,” *Proc. - 2020 IEEE 5th Int. Conf. Data Sci. Cyberspace, DSC 2020*, pp. 22–29, 2020, doi: 10.1109/DSC50466.2020.00012.

[14] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, “Stance detection with bidirectional conditional encoding,” 2016, doi: 10.18653/v1/d16-1084.

[15] S. M. Reddy, C. Suman, S. Saha, and P. Bhattacharyya, “A GRU-

based fake news prediction system: Working notes for UrduFake-FIRE 2020," *CEUR Workshop Proc.*, vol. 2826, pp. 464–468, 2020.

[16] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," Universiteit van Amsterdam, The Netherlands.

[17] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Vector Space," pp. 1–12.

[18] deryrahman, "Word2Vec Bahasa Indonesia," *https://github.com/*, 2019.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Trans. Assoc. Comput. Linguist.*, vol. 5, 2017, doi: 10.1162/tacl_a_00051.

[20] A. T. Mohan and D. V. Gaitonde, "A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks," *Cornell Univ.*, no. April, 2018.

[21] Alex Graves, "Supervised Sequence Labelling with Recurrent Neural Networks," *Springer, Berlin, Heidelb.*, 2012, doi: https://doi.org/10.1007/978-3-642-24797-2.

[22] K. Cho, "Learning Phrase Representations using RNN Encoder – Decoder for Statistical Machine Translation," *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process. (EMNLP), Doha, Qatar*, pp. 1724–1734, 2014.

[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014.

[24] Simeon Kostadinov, "Understanding GRU Networks," 2017.