

IDENTIFIKASI UJARAN KEBENCIAN PADA SOSIAL MEDIA BAHASA INGGRIS MENGGUNAKAN *RECURSIVE NEURAL NETWORK*

Jasman Pardede¹, Rangga Alfiansyah²

^{1,2} Program Studi Informatika, Institut Teknologi Nasional Bandung
Jl. Phh. Mustofa No.23, Neglasari, Kec. Cibeunying Kaler, Kota Bandung, Jawa Barat 40124
E-mail : jasman@itenas.ac.id¹

Abstrak

Penyebaran ujaran kebencian pada sosial media telah menjadi masalah serius karena menciptakan serangkaian konflik ras, etnis, orientasi seksual, kebangsaan, dan agama. Penelitian ini menggali dari unggahan ujaran kebencian di media sosial. Ujaran kebencian semakin meningkat di media sosial yang merendahkan individu atau kelompok banyak ditemukan. Mengidentifikasi ujaran kebencian secara tepat menjadi penting untuk menganalisis sentimen publik dari kelompok pengguna terhadap kelompok lain serta mencegah kegiatan yang tidak diinginkan. Dengan mengidentifikasi akar penyebabnya, pihak-pihak yang berwenang dan platform-platform digital dapat mengambil tindakan yang lebih efektif dalam memitigasi dampak negatifnya dan melindungi masyarakat dari ancaman yang mungkin timbul. Algoritma seperti *Support Vector Machine* dan *Deep Neural Networks*, terutama *Recursive Neural Network*, dapat mengidentifikasi ujaran kebencian. Dalam penelitian ini menerapkan *Recursive Neural Network* untuk mengidentifikasi ujaran kebencian. Performa sistem ini diukur berdasarkan nilai *precision*, *recall*, *accuracy*, dan *F-Measure*. Hasil eksperimen yang dilakukan terhadap data ujaran kebencian diperoleh nilai rata-rata *precision*, *recall*, *accuracy*, dan *F-Measure* masing-masing sebesar 0.78, 0.74, 0.76, dan 0.76. Hasil tersebut menunjukkan bahwa *Recursive Neural Networks* yang diusulkan memiliki performa yang cukup baik.

Kata kunci : Ujaran kebencian, Identifikasi, Sentimen Analisis, *Recursive Neural Network*, *Recurrent Neural Network*

Abstract

The spread of hate speech on social media has become a serious problem because it creates a series of conflicts over race, ethnicity, sexual orientation, nationality, and religion. This research explores hate speech posts on social media. Hate speech is increasing on social media that denigrates individuals or groups. Properly identifying hate speech is important to analyse the public sentiment of user groups towards other groups and prevent unwanted activities. By identifying the root cause, authorities and digital platforms can take more effective actions to mitigate the negative impact and protect the public from possible threats. Algorithms such as Support Vector Machine and Deep Neural Networks, especially Recursive Neural Network, can identify hate speech. This research applies Recursive Neural Network to identify hate speech. The performance of this system is measured based on the precision, recall, accuracy, and F-Measure values. The results of experiments conducted on hate speech data obtained average values of precision, recall, accuracy, and F-Measure of 0.78, 0.74, 0.76, and 0.76, respectively. These results show that the proposed Recursive Neural Networks perform quite well.

Keywords : Hate speech, Identification, Analysis Sentiment, *Recursive Neural Network*, *Recurrent Neural Network*

1. PENDAHULUAN

Media sosial adalah media daring yang mendukung interaksi sosial dan media sosial menggunakan teknologi berbasis web yang mengubah komunikasi menjadi dialog yang lebih interaktif antara pengirim dan penerima [1]. Dengan peningkatan besar dalam interaksi di jejaring sosial media, ada juga peningkatan kegiatan penyebaran kebencian. Ujaran Kebencian adalah "perkataan yang meremehkan orang atau grup berdasarkan beberapa karakteristik seperti ras, etnis, jenis kelamin, orientasi seksual, kebangsaan, agama,

atau karakteristik lainnya“ [2]. Mendeteksi ujaran kebencian semacam itu penting untuk menganalisis publik dari sekelompok pengguna terhadap kelompok lain, dan untuk mencegah konflik yang tidak diinginkan.

Menurut riset *Wearesosial Hootsuite* pada Januari 2019, pengguna media sosial di Indonesia mencapai 150 juta orang, yang merupakan 56% dari total populasi. Angka ini mengalami peningkatan 20% dari survei sebelumnya oleh Databoks pada tahun 2019 [3]. Data dari *Federal Bureau of Investigation (FBI)* mencatat lebih dari 7120 kasus ujaran kebencian selama tahun 2018 [4], termasuk serangan berdasarkan ras, agama, orientasi seksual, dan disabilitas. Direktorat Tindak Pidana Siber (Dittipidsiber) Bareskrim Polri juga melaporkan bahwa penyebaran konten provokatif merupakan laporan terbanyak kedua dengan 1769 kasus selama tahun 2019 [5].

Untuk mengidentifikasi ujaran kebencian di media sosial, algoritma sangatlah penting. Penelitian sebelumnya oleh [6] mencapai akurasi 61,667% dengan menggunakan *Support Vector Machine*, sementara [7] menjelaskan bahwa metode yang sering dipakai yaitu *Random Forest* dan *Gradient Boosted Decision Trees*. Metode *Deep Neural Network*, terutama *Recurrent Neural Networks*, adalah metode populer untuk Natural Language Processing karena kesesuaiannya untuk memproses teks yang panjang [8]. *Recursive Neural Network* merupakan salah satu model arsitektur dari *Recurrent Neural Network*. Model *Recursive Neural Network* merupakan salah satu model yang terbaik untuk analisis sentimen [9].

Penelitian [10] melakukan pendeteksian ujaran kebencian berbahasa Inggris pada media sosial. Mereka menggunakan metode *Logistic Regression*, *NaiveBayes*, *Decision Trees*, *Random Forest*, dan *Linear SVMs*. Sehingga penelitian ini akan mengusulkan akan menggunakan metode *Recursive Neural Network* untuk mendeteksi ujaran kebencian berbahasa Inggris.

2. METODOLOGI

2.1 Ujaran Kebencian

Ujaran Kebencian adalah bahasa yang bersifat menyerang dan mengandung kekerasan atau kebencian terhadap kelompok, berdasarkan karakteristik seperti penampilan fisik, agama, keturunan, etnis, orientasi seksual, identitas jenis kelamin atau lainnya, dan dapat terjadi dengan bahasa yang berbeda gaya, bahkan dalam bentuk kalimat halus atau hanya bercanda [7]. Jejaring sosial media juga berkontribusi untuk mengganggu masyarakat dalam berbagai cara. Seperti, ujaran kebencian melalui sosial media berkontribusi terhadap kegaduhan dalam kemasyarakatan [11].

2.2 Word Embeddings

Word2Vec adalah salah satu metode *Word Embeddings* untuk membuat representasi vektor dari kata-kata berdasarkan konteksnya. Model skip gram adalah salah satu variasi dari *Word2Vec* yang memprediksi kata-kata konteks dari sebuah kata target. Model ini dapat menghasilkan vektor kata yang menangkap makna semantik dan sintaktis dari kata-kata [12]. Rumus dasar untuk model *Word2Vec Skip-Gram* adalah sebagai berikut:

Misalkan sebuah kalimat berisi kata-kata $\omega_1, \omega_2, \omega_3, \dots, \omega_T$, dan ingin meminimalkan fungsi tujuan (*loss function*) dengan tujuan memaksimalkan probabilitas kemunculan kata-kata konteks ω_c dalam jangkauan sekitar kata input ω_t .

Fungsi tujuan dalam model *Skip-Gram*:

$$J = \frac{1}{T} \sum_{t=1}^T \sum_{c \leq j \leq c, j \neq 0} \log P(\omega_{t+j} | \omega_t) \quad (1)$$

Dimana:

T : panjang total korpus dalam jumlah kata.

c : ukuran jendela konteks.

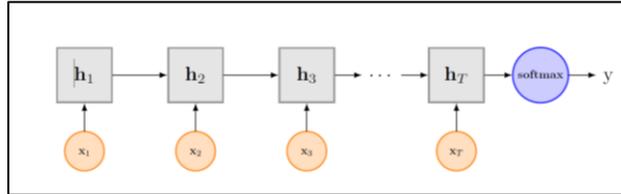
$P(\omega_{t+j} | \omega_t)$: probabilitas kemunculan kata konteks ω_{t+j} diberikan kata *input* ω_t . Ini dihitung menggunakan fungsi *softmax*:

$$P(\omega_{t+j} | \omega_t) = \frac{e^{\text{score}(\omega_{t+j}, \omega_t)}}{\sum_{w' \in \text{Vocabulary}} e^{\text{score}(w', \omega_t)}} \quad (2)$$

Nilai skor ($\omega_{t+j}|\omega_t$) dapat dihitung dengan *dot product* antara vektor representasi kata ω_{t+j} dan vektor representasi kata *input* ω_t

2.3 Recurrent Neural Network

Fitur mendasar dari *Recurrent Neural Network* (RNN) adalah jaringan yang pemrosesannya dipanggil berulang-ulang untuk memproses masukan yang biasanya adalah data sekuensial. Itu memungkinkan jaringan untuk melakukan pemrosesan temporal dan mempelajari urutan [13]. Sehingga RNN memiliki isi memori informasi dari hasil yang sebelumnya.



Gambar 1. Recurrent Neural Network [8]

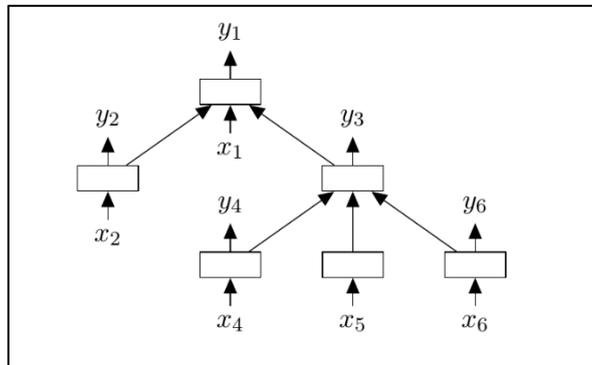
Setiap langkah dalam RNN menggunakan *input* sebelumnya

$$x_{(t-1)} \tag{3}$$

dan *input* x_t untuk menghasilkan y_t dan disimpan dalam memori baru h_t . Pada memori tersebut hasil rekaman yang telah disimpan kemudian dikalkulasikan sesuai memori sebelumnya.

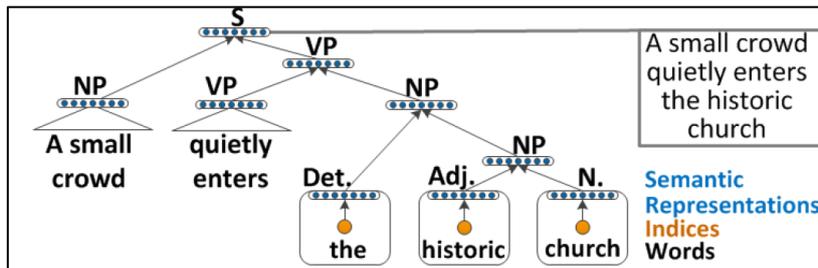
2.4. Recursive Neural Network

Recursive Neural Network (RecNN) adalah salah satu arsitektur dari *Recurrent Neural Network* (RNN). RecNN menggabungkan kata-kata untuk menyatakan sebagai kalimat dan menjadi metode terbaik untuk memprediksi sentimen pada sebuah kalimat [9].



Gambar 2. Model Recursive Neural Networks [14]

Recursive Neural Network membutuhkan struktur topologi untuk memodelkan sebuah kalimat contohnya seperti pohon pada Gambar 2. Struktur model pohon dapat memodelkan fitur-fitur dengan memroses informasi dari layer terbawah hingga layer tertinggi [15]. Ketika simpul paling bawah dikombinasikan dengan simpul atas, maka kombinasi informasi antara dua simpul bawah digabungkan dan direpresentasikan sebagai simpul di atasnya.



Gambar 3. Ilustrasi Recursive Neural Networks [16]

Satu bentuk umum dari RecNN adalah *Recursive Autoencoders*. Model ini bekerja dengan memetakan input ke representasi vektor-*latent*, lalu menggunakan representasi ini untuk menghasilkan *output* yang kemudian didekoderkan kembali ke dalam domain semula. Contohnya, terdapat pohon sintaksis dari kalimat, dapat menggunakan *Recursive Autoencoders* untuk melakukan pelatihan pemetaan dari *node-node* pohon menjadi representasi vektor, dan kemudian melakukan dekodingnya kembali untuk merekonstruksi pohon semula. Persamaan dasar yang digunakan dalam *Recursive Autoencoders* adalah:

1. *Encoder*:

$$h_i = f(W_{enc} \cdot [x_i, h_{i1}, h_{i2}]) \tag{4}$$

Dimana:

x_i : vektor fitur input untuk node i .

h_{i1}, h_{i2} : adalah vektor representasi dari anak-anak node i .

2. *Decoder*:

$$\hat{x}_i = g(W_{dec} \cdot h_i) \tag{5}$$

Dimana:

\hat{x}_i : rekonstruksi vektor dari node i .

g : fungsi aktivasi output.

W_{dec} : matriks bobot dekoder

2.5 Dataset

Data yang akan digunakan adalah dataset dari media sosial *Twitter* yang berjudul “*Hate speech offensive tweets*” diunduh dari *Kaggle* dan dibuat oleh [10], yang berjumlah 2258. Penelitian [10] menggunakan *logistic regression classifier* menghasilkan akurasi tertinggi 90,92% pada set pengujian. Selanjutnya, dataset yang tersedia akan dipecah menjadi perbandingan 80-10-10, dimana 80% akan digunakan sebagai data *training*, 10% sebagai data *validation*, dan 10% sebagai data *testing*. Dataset yang sedang diolah memiliki format dengan kolom kalimat dan label. Dalam kolom label terdapat kategori “ujaran kebencian” dan “bukan ujaran kebencian”. Label “0” digunakan untuk mengindikasikan informasi “ujaran kebencian”, sedangkan label “1” digunakan untuk menunjukkan informasi “bukan ujaran kebencian”.

Tabel 1. Contoh Dataset

No	Kalimat	Label
1	@AnUglyN*gga: this is ret*rded	0
2	"@MarkRoundtreeJr: LMFAOOOO I HATE BLACK PEOPLE https://t.co/RNvD2nLCCR" This is why there's black people and n*ggers	0
3	#Dutch farmers are white tr*sh.	0
4	" cancel that b*tch like Nino "	1
5	@AbrehamDadi no b*tch	1
6	@MrCatLoverx text me b*tch	1

2.6 Preprocessing

Sebelum membuat desain model RecNN, dilakukan tahap *preprocessing*, yang meliputi *case folding*, *punctuation removal*, *stopword removal*, dan *tokenization*. Tahap *preprocessing* dilakukan dengan langkah-langkah sebagai berikut.

1. Case Folding

Tujuan dari penerapan *case folding* adalah untuk menstandarkan semua variasi karakter dalam teks kalimat, menyederhanakan penghapusan karakter atau kata tertentu yang tidak diinginkan. Dalam penelitian ini, semua huruf dalam dokumen diubah menjadi huruf kecil, hanya menerima rentang huruf 'a' hingga 'z'.

2. Punctuation Removal

Dalam langkah ini, tanda baca yang dimaksud mencakup simbol seperti tanda tanya (?), seru (!), koma (,), garis miring (/), tanda sama dengan (=), tanda tambah (+), tanda minus (-), garis miring terbalik (\), lebih besar (>), lebih kecil (<), titik koma (;), tanda kutip (“”), tanda kurung (()), tanda kurung kurawal ({}), tanda kurung siku ([]), dan lainnya. Penghapusan simbol-simbol ini dilakukan karena selama proses *training*, tanda baca diabaikan sehingga menghilangkan tanda baca ini akan menyederhanakan proses *training*.

3. Tokenization

Tokenization berfungsi untuk memecah teks menjadi kata-kata atau mengubah seluruh kalimat pada data menjadi kata-kata atau *token*, sehingga memudahkan mesin untuk memproses data.

Tabel 2 merupakan contoh dataset yang telah melalui tahap preprocessing.

Tabel 2. Contoh Dataset Yang Telah Di-Preprocessing

No	Kalimat	Label
1	this is ret*rded	0
2	i hate black people this is why there is black people and n*ggers	0
3	farmers are white tr*sh	0
4	cancel that bitch like nino	1
5	no b*tch	1
6	text me b*tch	1

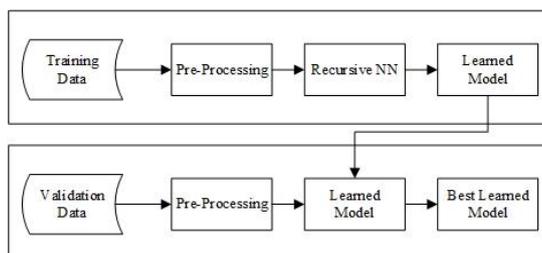
2.7 Training

Langkah pertama dalam proses *training* melibatkan pembuatan model menggunakan arsitektur RecNN (*Recursive Neural Network*). Langkah ini dimulai dengan mengunduh dataset ujaran kebencian dari situs web *kaggle.com*. Setelah itu, data tersebut telah diberi label untuk membedakan ujaran kebencian dan bukan ujaran kebencian.

Selanjutnya, dilakukan tahap *preprocessing* pada data tersebut. Langkah-langkah dalam *preprocessing* termasuk meratakan huruf (*case folding*), menghapus tanda baca (*punctuation removal*), dan membagi teks menjadi unit-unit yang lebih kecil (*tokenization*).

Setelah *preprocessing* selesai, langkah berikutnya adalah memulai proses pembuatan model RecNN. Langkah pertama dalam proses ini adalah mengubah teks menjadi representasi numerik yang disebut vektor penanaman kata (*word embedding*). Langkah ini bertujuan untuk mengubah data yang telah diproses sebelumnya menjadi bentuk *tensor* tiga dimensi, yang akan menjadi masukan bagi model RecNN. Setelah mendapatkan *tensor* tiga dimensi, langkah selanjutnya adalah menjalankan data melalui model RecNN.

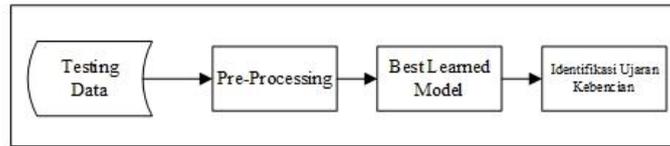
Setelah proses *training* selesai. Penting untuk mencatat bahwa untuk menghasilkan model yang valid, kinerja model yang telah dilatih dievaluasi menggunakan data validasi.



Gambar 4. Tahapan Proses Training

2.8 Testing

Pada fase ini, setelah melalui proses *training*, evaluasi kinerja model dilakukan dengan *testing* model terbaik menggunakan 10% data uji dari seluruh kumpulan data.



Gambar 5. Tahapan Proses Seluruh Alur Kerja Sistem

2.8 Evaluasi

Evaluasi model RecNN dilakukan dengan sekali uji dengan memanfaatkan *confusion matrix* dan mengukur kinerjanya menggunakan *precision*, *recall*, *accuracy*, dan *f-measure*. *Confusion matrix* yang digunakan dapat ditemukan pada Gambar 12. Untuk menghitung nilai *precision*, *recall*, *accuracy*, dan *f-measure* [17] digunakan rumus yang tercantum dalam Persamaan (6) sampai dengan (9).

$$precision = \frac{TP}{TP + FP} \tag{6}$$

$$recall = \frac{TP}{TP + FN} \tag{7}$$

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{8}$$

$$F - Measure = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{9}$$

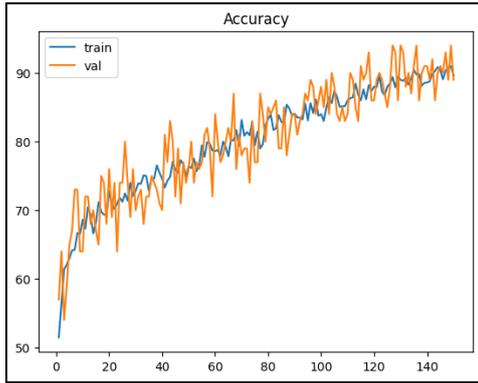
Dimana :

- TP : Dikategorikan ujaran kebencian; sebenarnya memang ujaran kebencian.
- TN : Dikategorikan bukan ujaran kebencian; sebenarnya memang bukan ujaran kebencian.
- FP : Dikategorikan ujaran kebencian namun sebenarnya bukan ujaran kebencian.
- FN : Dikategorikan bukan ujaran kebencian namun sebenarnya ujaran kebencian.

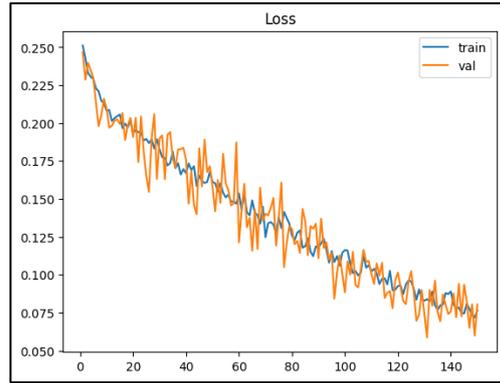
3. HASIL DAN PEMBAHASAAN

Penelitian ini menggunakan perangkat keras dengan konfigurasi berbeda. Proses *training* dilakukan pada sebuah mesin dengan processor NVIDIA T4 berkecepatan 585MHz, RAM sebesar 13GB, dan kapasitas penyimpanan 78GB yang disediakan melalui platform *Google Collaboratory*. Sementara itu, *testing* dilakukan dengan menggunakan mesin yang dilengkapi processor Intel Core i5-8300H berkecepatan 2.3GHz, RAM sebesar 8GB, dan kapasitas penyimpanan 1TB. Framework yang digunakan dalam penelitian ini adalah Tensorflow.

Tujuan dari penelitian ini adalah untuk mengidentifikasi probabilitas tertinggi berdasarkan kategori ujaran kebencian atau bukan kebencian dengan menerapkan metode RecNN. Data *training* dibagi menjadi perbandingan 80% data *training*, 10% data *validation*, dan 10% data *testing*. Model dikembangkan dengan menggunakan 150 *epoch*, ukuran *batch* sebesar 100, dan *layer* RecNN sejumlah 512. Dalam penelitian ini, dilakukan serangkaian eksperimen menggunakan teknik *dropout* dengan nilai yang berbeda, yaitu 0.20, 0.25, 0.3, 0.35, 0.4, 0.45, dan 0.5, dengan pengaturan *epoch* dan ukuran *batch* yang sama. Gambar-gambar 6 hingga 12 menampilkan sumbu (x) yang mewakili informasi epoch dari 0 hingga 150, sementara sumbu (y) menunjukkan nilai *loss* dan *accuracy* yang terkait. Kurva berwarna biru mewakili tren data *training*, sedangkan garis berwarna oranye menggambarkan tren data *validation*. Gambar-gambar 6 sampai 12 memvisualisasikan grafik *accuracy* dan *loss* dengan nilai *dropout* berturut-turut 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, dan 0.5.

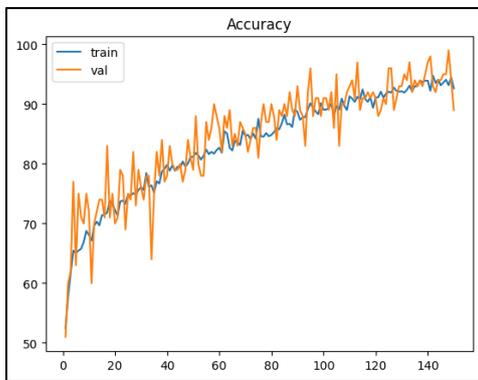


(a) Nilai Accuracy

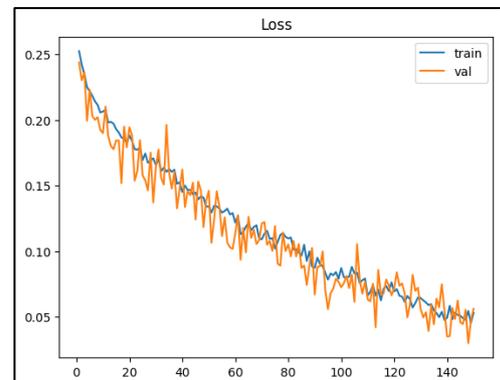


(b) Nilai Loss

Gambar 6. Hasil Accuracy Dan Loss Dengan Nilai Dropout 0,2.

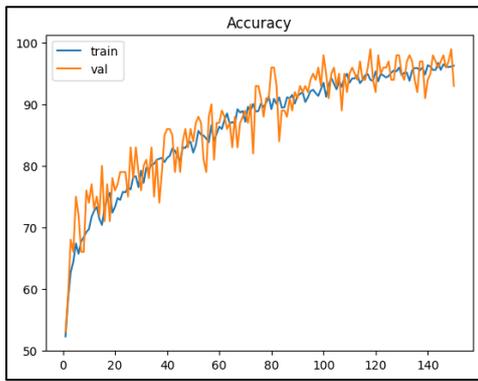


(a) Nilai Accuracy

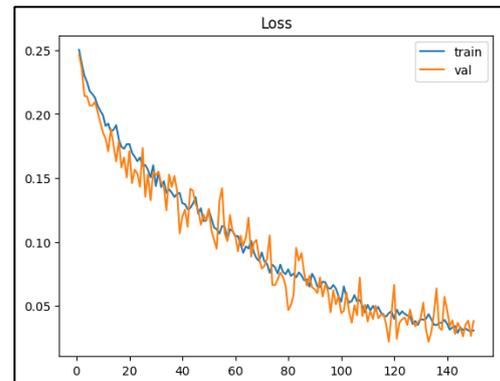


(b) Nilai Loss

Gambar 7. Hasil Accuracy Dan Loss Dengan Nilai Dropout 0,25.

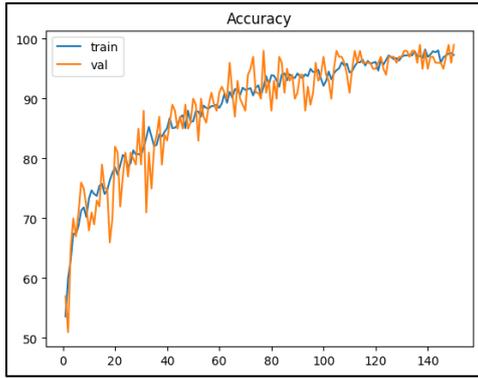


(a) Nilai Accuracy

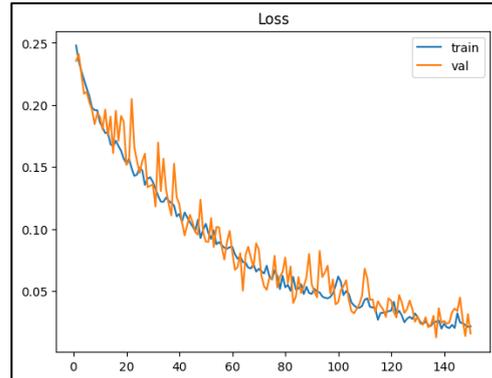


(b) Nilai Loss

Gambar 8. Hasil Accuracy Dan Loss Dengan Nilai Dropout 0,3.

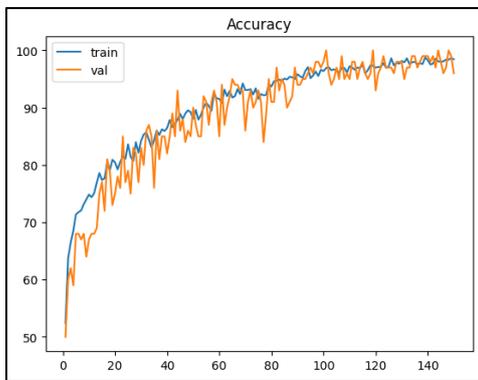


(a) Nilai Accuracy

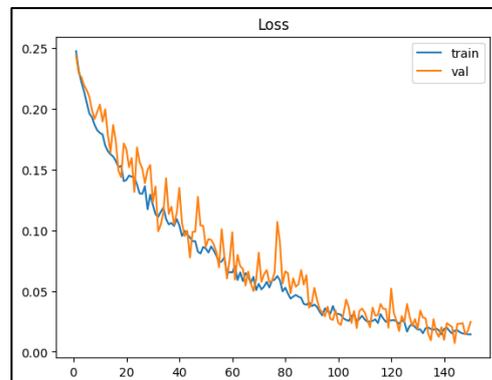


(b) Nilai Loss

Gambar 9. Hasil Accuracy Dan Loss Dengan Nilai Dropout 0,35.

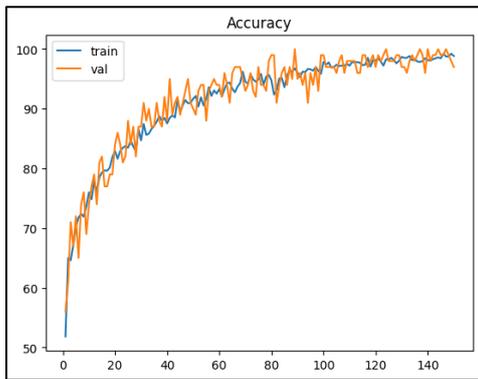


(a) Nilai Accuracy

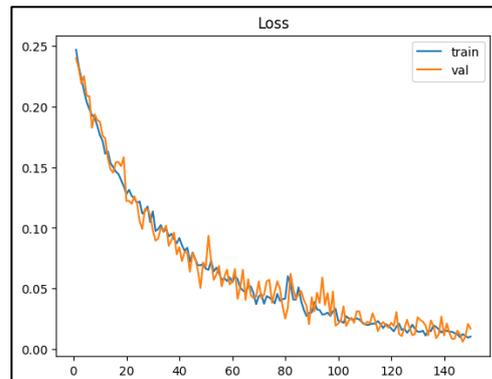


(b) Nilai Loss

Gambar 10. Hasil Accuracy Dan Loss Dengan Nilai Dropout 0,4.

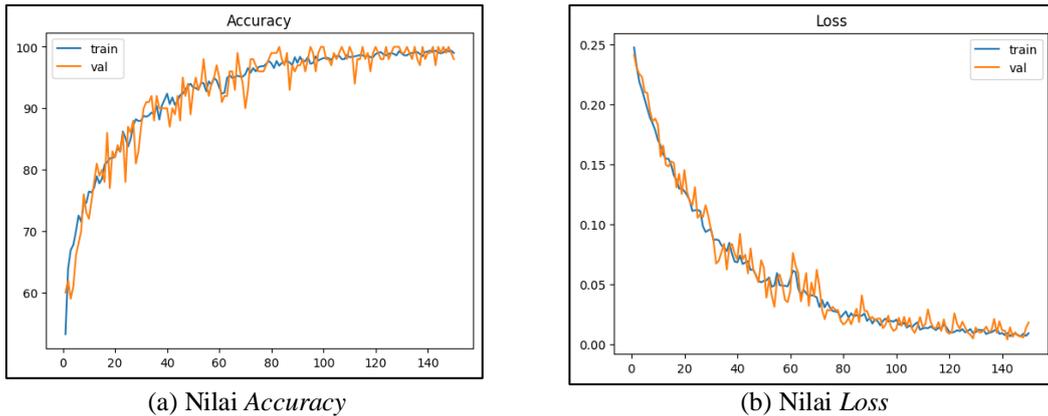


(a) Nilai Accuracy



(b) Nilai Loss

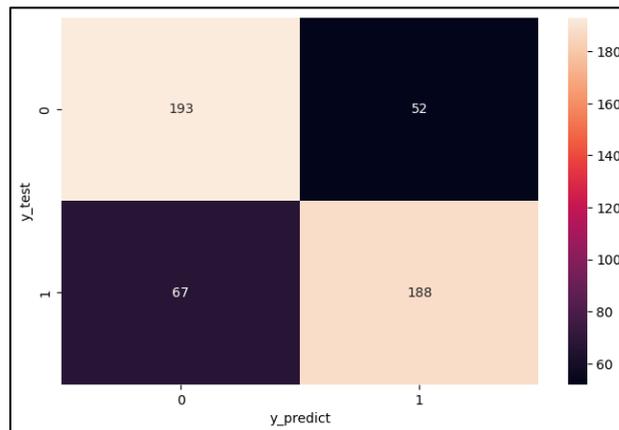
Gambar 11. Hasil Accuracy Dan Loss Dengan Nilai Dropout 0,45.



Gambar 12. Hasil Accuracy Dan Loss Dengan Nilai Dropout 0,5.

Pada Gambar 6 hingga Gambar 12, terlihat bagaimana perubahan *loss* dan *accuracy* selama proses *accuracy* dengan variasi nilai *dropout*, yaitu dari 0.2 hingga 0.5. Dari analisis grafik-grafik tersebut, eksperimen dengan nilai *dropout* 0.5 menunjukkan kinerja yang lebih baik dibandingkan dengan eksperimen menggunakan nilai *dropout* 0.2 hingga 0.4.

Setelah memperoleh model RecNN dengan kinerja terbaik dari proses *training*, model tersebut diuji dengan menggunakan data *testing* yang memiliki ukuran 10% dari total data. Untuk mengevaluasi kinerja model yang dihasilkan, digunakan *confusion matrix* dengan parameter *precision*, *recall*, *accuracy*, dan skor *f-measure*. Hasil dari evaluasi ini ditampilkan dalam Gambar 12 sebagai hasil dari *confusion matrix*.



Gambar 13. Confusion Matrix

Berdasarkan Gambar 13, terdapat total 590 data *testing* yang dievaluasi. Dari data tersebut, 193 data *testing* yang sebenarnya adalah ujaran kebencian telah diprediksi dengan benar sebagai ujaran kebencian (TP), dan 52 data *testing* yang sebenarnya adalah bukan ujaran kebencian namun diprediksi sebagai ujaran kebencian (FP). Selanjutnya, terdapat 188 data *testing* yang benar-benar merupakan bukan ujaran kebencian dan telah diprediksi dengan benar sebagai bukan ujaran kebencian (TN), serta 67 data *testing* yang seharusnya adalah ujaran kebencian namun dideteksi sebagai bukan ujaran kebencian (FN). Oleh karena itu, setelah menghitung 193 TP, 52 FP, 67 FN, dan 188 TN, ditemukan bahwa *precision*, *recall*, *accuracy*, dan skor *f-measure* memiliki nilai masing-masing sebesar 0,78, 0,74, 0,76, dan 0,76.

4. PENUTUP

Dalam penelitian ini, metode RecNN telah diaplikasikan untuk mengidentifikasi ujaran kebencian dalam bahasa Inggris di media sosial. Metode yang diajukan mampu mengidentifikasi ujaran kebencian dengan nilai rata-rata *precision*, *recall*, *accuracy*, dan *f-measure* mencapai 0,78, 0,74, 0,76, dan 0,76. Temuan penelitian ini menunjukkan bahwa metode RecNN memberikan hasil yang cukup baik.

Langkah berikutnya dalam penelitian diharapkan mampu meningkatkan performa dengan memperluas kosa kata melalui pemilihan korpus yang lebih beragam, sehingga dapat menghasilkan variasi kosa kata yang lebih kaya.

DAFTAR PUSTAKA

- [1] E. Susanti, "Glosarium Kosakata Bahasa Indonesia Dalam Ragam Media Sosial," *DIALEKTIKA: Jurnal Bahasa, Sastra, dan Pendidikan Bahasa dan Sastra Indonesia*, hal. 229-250, 2016.
- [2] I. Alfina, D. Sigmawaty, F. Nurhidayati dan A. N. Hidayanto, "Utilizing Hashtags for Sentiment Analysis of Tweets in The Political Domain," *Proceedings of the 9th International Conference on Machine Learning and Computing*, hal. 43-47, 2017.
- [3] Databoks, "Berapa Pengguna Media Sosial Indonesia?," Januari 2019. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2019/02/08/berapa-pengguna-media-sosial-indonesia>.
- [4] FBI, "Hate Crime Statistics," U.S. Department of Justice, Washington, DC, 2018.
- [5] B. Polri, "Statistik," 2019. [Online]. Available: <https://patrolisiber.id/statistic>.
- [6] D. P. N. Lyrawati, "Deteksi Ujaran Kebencian Pada Twitter Menjelang Pilpres 2019 Dengan Machine Learning," *Jurnal Ilmiah Matematika*, vol. 7, no. 3, 2019.
- [7] P. Fortuna dan S. Nunes, "A Survey on Automatic Detection of Hate Speech in Text," *ACM Computing Surveys*, hal. 1-30, 2018.
- [8] P. Liu, X. Qiu dan X. Huang, "Recurrent Neural Network for Text Classification with Multi-Task Learning," *International Joint Conference on Artificial Intelligence*, 2016.
- [9] T. H. Nguyen dan K. Shirai, "PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, hal. 2509-2514, 2015.
- [10] T. Davidson, D. Warmesley, M. Macy dan I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," *Proceedings of the 11th International AAAI Conference on Web and Social Media*, hal. 512-515, 2017.
- [11] Z. Mossie dan J.-H. Wang, "Vulnerable community identification using hate speech detection," *Information Processing and Management*, vol. 57, no. 3, 2019.
- [12] P. T. Hung dan K. Yamanishi, "Word2vec Skip-Gram Dimensionality Selection via Sequential Normalized Maximum Likelihood," *Entropy*, vol. 23, no. 8, hal. 997, 2021.
- [13] J. A. Bullinaria, "Recurrent Neural Networks," 2015. [Online]. Available: <http://www.cs.bham.ac.uk>.
- [14] K. S. Tai, R. Socher dan C. D. Manning, "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, hal. 1556-1566, 2015.
- [15] X. Chen, X. Qiu, C. Zhu, S. Wu dan X. Huang, "Sentence Modeling with Gated Recursive Neural Network," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, hal. 793-798, 2015.
- [16] R. Socher, C. C.-Y. Lin, A. Y. Ng dan C. D. Manning, "Parsing Natural Scenes and Natural Language with Recursive Neural Networks," *Proceedings of the 28th international conference on machine learning (ICML-11)*, hal. 129-136, 2011.
- [17] J. Pardede dan M. G. Husada, "Comparison of VSM, GVSM, and LSI in Information Retrieval For Indonesian Text," *Jurnal Teknologi Malaysia*, vol. 78, no. 2180-3722, hal. 5-6, 2015.