

PERBANDINGAN METODE LSTM DAN GRU (RNN) UNTUK KLASIFIKASI BERITA PALSU BERBAHASA INDONESIA

COMPARISON OF LSTM AND GRU (RNN) METHODS FOR FAKE NEWS CLASSIFICATION IN INDONESIAN

Aini Hanifa^{*1}, Sugih Ahmad Fauzan², Muhammad Hikal³, Muhammad Bahrul Ashfiya⁴

*Email: ainihanifa@unsoed.ac.id

^{1,2,3,4} Jurusan Informatika, Fakultas Teknik, Universitas Jenderal Soedirman, Purwokerto

Abstrak— Berita telah menjadi konsumsi masyarakat setiap harinya, namun tidak semua berita yang beredar merupakan berita yang valid kebenarannya. Berita palsu dapat menggiring opini publik, dan berisiko terhadap keselamatan bangsa. Oleh karena itu diperlukan klasifikasi berita palsu untuk dapat meredamkan berita palsu yang beredar pada masyarakat. Penelitian sebelumnya telah dilakukan klasifikasi berita palsu menggunakan model *Recurrent Neural Network* (RNN) yaitu *Long Short-Term Memories* (LSTM), dengan nilai F1 sebesar 0,24 dan menyarankan untuk mencari parameter model sistem yang tepat agar dihasilkan kinerja model yang lebih baik. Maka pada penelitian ini dilakukan klasifikasi berita palsu berbahasa Indonesia dengan menggunakan perbandingan model *Recurrent Neural Network* yaitu LSTM dan *Gated Recurrent Unit* (GRU), serta mencari parameter terbaik untuk menghasilkan hasil kinerja klasifikasi paling optimal. Data yang digunakan merupakan berita mengenai kejadian yang terjadi di Indonesia dan berbahasa Indonesia. Didapatkan nilai parameter *epochs*: 15, fungsi *optimizer gradient descent* yaitu *rmsprop*, dan *batch_size*: 64 untuk mendapatkan kinerja optimal dalam klasifikasi berita palsu menggunakan metode LSTM. Nilai akurasi yang dihasilkan oleh model yaitu 73% untuk metode LSTM dan 64% dengan menggunakan metode GRU.

Kata kunci — berita palsu, *Recurrent Neural Network*, *Long Short-Term Memories*, *Gated Recurrent Unit*, parameter.

Abstract— News has become public consumption every day, but not all news is valid news. Fake news can lead to public opinion, and put the nation's safety at risk. Therefore it is necessary to classify fake news in order to reduce the spread of fake news in the community. Previous research has carried out the classification of fake news using the *Recurrent Neural Network* (RNN) model, namely *Long Short-Term Memories* (LSTM), where it just produces an F1 of 0.24, and suggested finding the right system model parameters in order to produce a better model performance. So in this study a classification of fake news in Indonesian was carried out using a comparison of the *Recurrent Neural Network* model, namely LSTM and *Gated Recurrent Unit* (GRU), along with looking for the best parameters to produce the most optimal classification performance results. The data used is news about events that have occurred in Indonesia and news in Indonesian. From the resulting experiment, the parameter value of *epochs*: 15, the *gradient descent optimizer function*: *rmsprop*, and *batch_size*: 64 to obtain optimal performance in the classification of fake news using the LSTM method. The accuracy value generated by the model is 73% using the LSTM method and 64% using the GRU method.

Keywords — fake news, *Recurrent Neural Network*, *Long Short-Term Memories*, *Gated Recurrent Unit*, parameter.

I. PENDAHULUAN

Berita palsu yang beredar di kalangan masyarakat khususnya di Indonesia kini kian meresahkan. Sebagian masyarakat mudah percaya dengan berita yang beredar, kemudian menyebarkan berita yang belum tentu benar kebenarannya. Berita palsu dapat menyebabkan perpecahan dalam negeri,

terutama di kalangan masyarakat dan juga dapat berbahaya. Berita palsu juga dapat menggiring opini masyarakat, sehingga sesuatu yang benar menjadi salah, begitu pula sebaliknya. Seperti pada saat kondisi pandemi saat ini, berita yang salah mengenai covid-19 sangat berisiko terhadap keselamatan bangsa. Oleh karena itu terdapat menu tersendiri

pada situs resmi satuan tugas penanganan covid-19 Indonesia mengenai klasifikasi berita palsu yaitu pada laman <https://covid19.go.id/p/hoax-buster>. Beberapa situs terkemuka telah membantu pemerintah untuk meredamkan berita palsu, seperti aplikasi perpesanan *Line* yang setiap sepekan mengeluarkan daftar berita palsu apa saja yang beredar pada pekan tersebut.

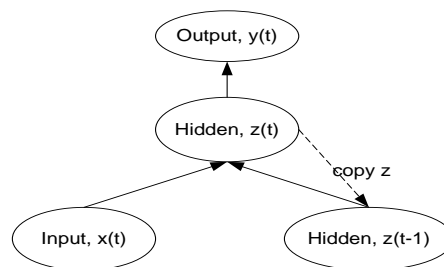
Berdasarkan permasalahan tersebut, dalam beberapa tahun terakhir, terdapat sejumlah penelitian mengenai klasifikasi berita palsu, baik berita berbahasa Indonesia maupun asing. Beberapa metode yang digunakan yaitu *Naïve Bayes* [1], *Ensemble Learning* [2], *SVM* dan *SGD* [3]. Rusli *et al.* [4] telah melakukan identifikasi berita palsu berbahasa Indonesia dengan menggunakan metode *Multi-Layer Perceptron* (MLP) dan menyarankan untuk penelitian selanjutnya menggunakan metode *embedding* atau metode klasifikasi *deep learning* untuk meningkatkan akurasi serta dapat menangani permasalahan *imbalance dataset*. Kemudian [5] melakukan perbandingan dari metode *Multilayer Perceptron*, *Naïve Bayes*, *SVM*, *Random Forest*, dan *Decision Tree*; dihasilkan metode *Random Forest* menunjukkan kinerja akurasi terbaik. Metode *Deep Learning* telah diterapkan oleh [6] dalam klasifikasi berita palsu berbahasa Indonesia, yaitu dengan metode *Long Short-Term Memory* (LSTM). Untuk klasifikasi berita palsu berbahasa asing, metode LSTM dapat dengan baik mengklasifikasikan data *balance* maupun *imbalance* dataset [7]. [8] juga telah menggunakan LSTM namun tidak menemukan hasil kinerja klasifikasi yang cukup baik yaitu dengan nilai F1 sebesar 0,24 dan menyarankan penelitian selanjutnya untuk mencari parameter model sistem yang tepat agar dihasilkan kinerja model yang lebih baik.

Berangkat dari permasalahan dan solusi yang ditawarkan di atas, maka pada penelitian ini dilakukan klasifikasi berita palsu berbahasa Indonesia dengan menggunakan perbandingan model *Recurrent Neural Network* yaitu *Long Short-Term Memories* (LSTM) dan *Gated Recurrent Unit* (GRU), serta mencari parameter terbaik untuk menghasilkan hasil kinerja klasifikasi paling optimal. Arsitektur metode *Recurrent Neural Networks* (RNN) merupakan bentuk arsitektur yang dapat memproses *sequential data*, sehingga tepat diterapkan dalam klasifikasi data teks.

II. TINJAUAN PUSTAKA

A. Recurrent Neural Network

Recurrent Neural Network (RNN) merupakan pengembangan dari algoritma yang sudah ada sebelumnya, yaitu *Artificial Neural Network*. Perbedaan terdapat pada RNN yang lebih difokuskan untuk data yang berupa *sequence* seperti kata ataupun kalimat. Pada RNN terdapat memori atau ingatan (*feedback loop*) yang dapat mengenali informasi dari masa lalu dalam proses pembelajaran modelnya. RNN sering diterapkan pada klasifikasi gambar (*image classification*), keterangan gambar (*image captioning*), pengenalan suara (*speech recognition*), prediksi saham (*stock prediction*), pemrosesan bahasa alami (*Natural Language Processing/NLP*), terjemahan mesin (*machine translation*), pemodelan bahasa tingkat karakter (*character-level language modeling*), dan rekayasa keuangan (*financial engineering*) [9] [10].



Gambar-1. Proses *Recurrent Neural Network* [11]

Berdasarkan Gambar-1, dapat dilihat bahwa secara garis besar proses *Recurrent Neural Network* adalah:

1. Menyalin input dari waktu t ke unit input.
2. Menghitung aktivasi *hidden unit* menggunakan *net input* dari *unit input* dan dari *copy layer*.
3. Menghitung unit aktivasi *output*.
4. Menyalin *hidden unit* yang baru ke *copy layer*.

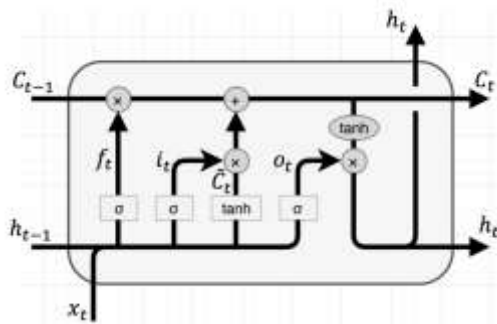
B. Vanishing Gradient Problem

Karena *Recurrent Neural Network* bekerja dalam suatu intens waktu (t), maka data yang telah diproses dalam intens waktu sebelumnya akan berpengaruh terhadap data yang akan diproses pada intens selanjutnya. Penghitungan *cost function* dan penyesuaian *weight* (W) dilakukan setiap intens waktu. Jika data akan berpindah dari $t-1$ ke t , maka perlu dikalikan dengan W , dan data akan terus melakukan proses tersebut jika *sequence data* yang dimiliki sangat panjang. Nilai W sangatlah kecil,

rentang nilainya antara 0,1-0,001, jika proses perkalian dengan W terjadi secara berulang-ulang, maka *gradient* yang dimiliki akan semakin mengecil. Semakin kecil *gradient* yang dimiliki maka semakin sulit untuk bisa mencapai *cost function*, sehingga masalah ini biasa disebut dengan *Vanishing Gradient Problem*. Dengan nilai *gradient* yang sangat kecil, maka pengaruhnya tidak akan signifikan pada model yang dibuat, bahkan jika *gradient* sudah mencapai *level vanishing*, maka model tidak akan dapat melakukan *learning* kembali, atau lebih tepatnya *cost function* dan *weight* tidak bisa diperbarui kembali, karena nilai yang sudah terlalu kecil. Jadi pada proses *learning* data yang memiliki *sequence* yang besar, diperlukan jaringan lain untuk bisa mengatasi masalah ini. Oleh karena itu, pada penelitian ini digunakan *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU).

a. Long Short-Term Memory

LSTM merupakan salah satu modifikasi dari *Recurrent Neural Network* (RNN) dengan menambahkan *memory cell* yang dapat menyimpan informasi untuk jangka panjang [11]. LSTM dibentuk untuk dapat menangani permasalahan *vanishing gradient* pada RNN saat memproses data *sequential* yang panjang. Permasalahan *vanishing gradient* tersebut berakibat RNN tidak mampu dalam menangkap *long term dependencies* [12], sehingga akurasi yang dihasilkan pada prediksi menggunakan metode RNN berkurang [13].



Gambar-2. Arsitektur LSTM

Pada Gambar-2 dapat dilihat bahwa LSTM mempunyai 3 *gate* yaitu *input gate*, *forget gate* dan *output gate*. Proses komputasi pada LSTM dilakukan secara bertahapan, yaitu sebagai berikut [14]:

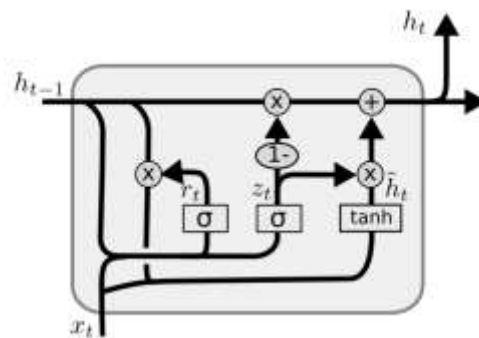
- **Input gate** memutuskan nilai mana yang akan diperbarui.
- **Forget gate** memutuskan informasi apa yang akan dibuang dari konteks C_{t-1} .
- **Output gate** memutuskan bagian mana saja dari konteks yang akan dihasilkan. Semua

gate memiliki dimensi yang sama dengan *dimension hidden state* sebagai ukuran untuk *hidden state*.

b. Gated Recurrent Unit

GRU memiliki kelebihan dibandingkan dengan LSTM yaitu komputasinya yang lebih sederhana, namun akurasi yang dihasilkan setara dan masih cukup efektif untuk menangani permasalahan *Vanishing Gradient* [15].

Pada Gambar-3 dapat dilihat bahwa GRU memiliki dua *gate*, *reset gate* r dan *update gate* z . *Reset gate* menentukan bagaimana cara menggabungkan input yang baru dengan memori sebelumnya. Sedangkan *update gate* menentukan seberapa banyak memori sebelumnya yang harus disimpan.

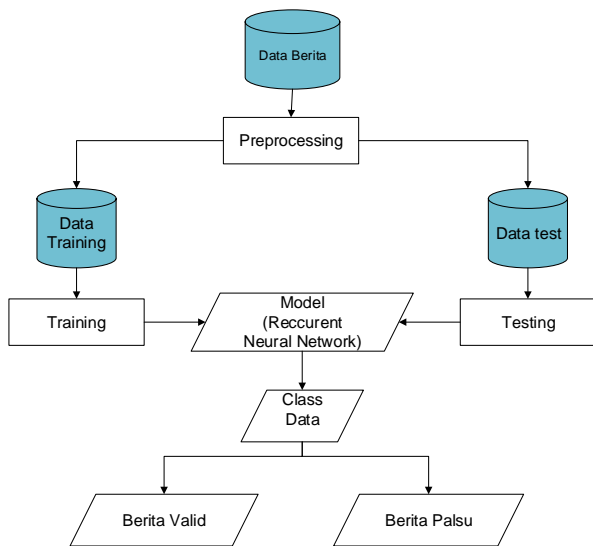


Gambar-3. Arsitektur GRU

Jika dilakukan pengaturan semua *reset* menjadi 1 dan *update gate* menjadi 0 maka akan digunakan model RNN yang standar. Ide dasar dari menggunakan mekanisme *gate* adalah untuk mempelajari *long-term dependencies*.

III. METODE

Pada penelitian ini, pembangunan model *machine learning* menggunakan arsitektur *Recurrent Neural Network*. Namun untuk mengatasi *Vanishing Gradient Problem* digunakan network *Long-Short Term Memory* dan *Gated Recurrent Unit*. Data dikumpulkan dari berbagai portal berita *on-line* di Indonesia, seperti; Detik, Tribun, Liputan 6, dan sebagainya. Lalu berita palsu disadur dari portal berita *turnbackhoax.id*. Total berita yang didapatkan berjumlah 600 berita, dengan sebaran 372 berita valid dan 228 berita palsu. Selanjutnya semua berita disimpan dalam 1 file dengan format *.csv* untuk selanjutnya dilakukan *text-preprocessing*. Secara umum, proses yang dilakukan oleh sistem dapat dilihat pada Gambar-4.



Gambar-4. Gambaran umum sistem

Sebelum menuju proses klasifikasi, dibutuhkan *pre-processing* yang tepat untuk menyajikan data yang sesuai untuk dianalisis. Berdasarkan data yang tersedia dan telah melalui *pre-processing* kemudian diolah membentuk sebuah model, dari model yang dibentuk kemudian didapat klasifikasi berita valid atau berita palsu. Proses *pre-processing* yang dilakukan antara lain, penghapusan *stopword*, tanda baca dan angka, *tokenizing* dan juga *padding*.

a) *Stemming*

Stemming yaitu proses mengubah kata menjadi bentuk kata dasarnya. Metode ini menyesuaikan struktur bahasa yang digunakan. Misalnya pada struktur Bahasa Indonesia, kata ‘pertolongan’ menjadi ‘tolong’. Proses ini diperlukan untuk menghindari mispersepsi kata, kata yang seolah berbeda ternyata memiliki kata dasar yang sama.

b) *Tokenizing*

Pada tahap ini dilakukan pemisahan kata-kata. Setiap potongan kata disebut dengan token atau term.

c) Penghapusan *stopword*, tanda baca dan angka

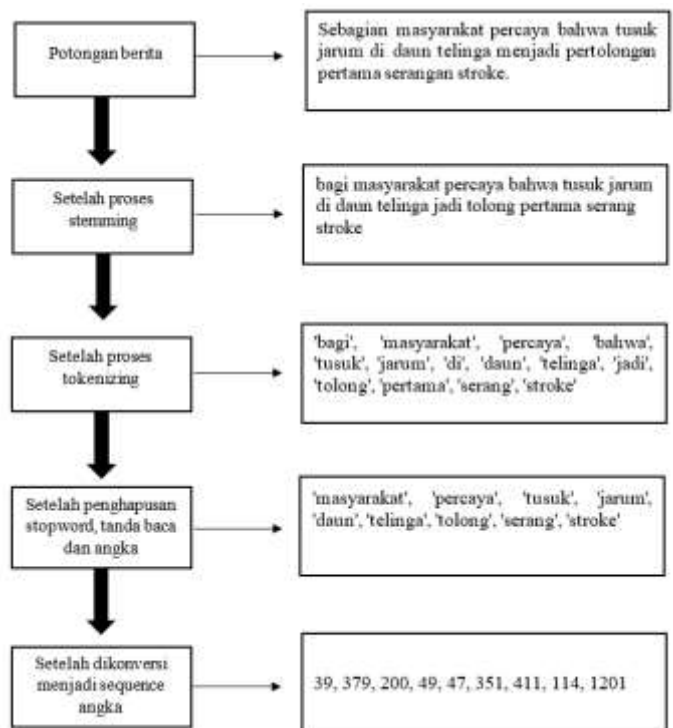
Proses *filter stopwords* dilakukan dengan menghapus kata-kata yang sering muncul namun tidak berpengaruh dalam pembentukan model. Kata yang termasuk *stopwords* yaitu seperti kata tanya, kata penunjuk waktu, dan tanda baca. Dalam proses ini menggunakan bantuan *library nltk*, untuk bisa mendapatkan list *stopword* dalam bahasa Indonesia. Berita akan di-*split* per kata dengan bantuan *library gensim*. Lalu setiap kata akan dicek apakah ada kata yang termasuk *stopword*, jika ya maka kata tersebut akan dihapus. Dengan menggunakan *gensim* maka tanda baca, simbol dan angka akan terhapus.

d) *Converting ke number sequence*

Agar data bisa di-*learn* oleh model *machine learning*, maka harus diubah menjadi angka, karena data merupakan berita atau teks atau berupa data *sequence*. Proses perubahan dari *text* ke *integer/* angka dilakukan per-kata, sebelum diterapkan pada algoritma *Recurrent Neural Network*. Proses perubahan ke *number sequence* menggunakan *method tokenizer texts_to_sequences*.

e) *Padding*

Proses *padding* adalah proses yang dilakukan untuk menyamakan panjang *sequence* pada tiap data. Proses ini dilakukan untuk bisa mempermudah model dalam memahami tiap data, karena panjang data disajikan dalam panjang *sequence* yang sama. Gambar-5 menunjukkan contoh potongan berita yang sudah melalui proses *text-preprocessing*.



Gambar-5. Alur proses *text-preprocessing*

Setelah melakukan *text-preprocessing*, data lalu di-*fit* ke model yang telah dirancang. Rancangan model *Recurrent Neural Network* tersebut dapat dilihat pada Gambar-6.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 240)	2149920
bidirectional (Bidirectional)	(None, 256)	377856
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 1)	129

Total params: 2,560,801
 Trainable params: 2,560,801
 Non-trainable params: 0

Gambar-6. Rancangan model *Recurrent Neural Network*

Dari Gambar-6 dapat dilihat beberapa *layer* yang dimiliki, yaitu :

1. *Embedding Layer*

Embedding layer disini berfungsi untuk menjadi *input layer*, dimana banyaknya data yang akan diinput adalah total kata yang ada pada dataset berita yang telah melalui *pre-processing*. Dan untuk output dimensionnya sebesar 240.

2. *Bidirectional Layer*

Pada *bidirectional layer* akan menggunakan *Network Long-Short Term Memory* dan *Gated Recurrent Unit*.

3. *Hidden Layer*

Pada *hidden layer* ini memiliki *input dimension* sebesar 128, pada *layer* ini menggunakan fungsi aktivasi ReLU (*Rectifier Linear Unit*).

4. *Output Layer*

Pada *output layer* menggunakan fungsi aktivasi Sigmoid.

IV. HASIL DAN PEMBAHASAN

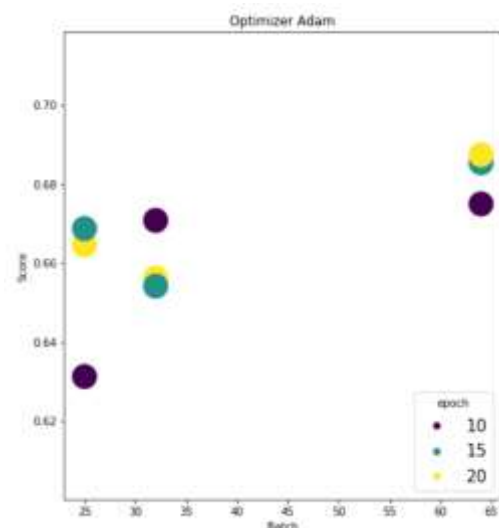
Setelah data di-*feed* ke model *Recurrent Neural Network* yang telah dirancang, dengan menggunakan library *accuracy_score*, Model *LSTM-RNN* mendapatkan nilai akurasi sebesar 0,38333333336, lalu model *GRU-RNN* mendapatkan nilai akurasi sebesar 0,38333333336. Hasil yang didapatkan masih kurang memuaskan, karena nilai yang didapatkan masih sangat rendah. Oleh karena itu dilakukan *tuning hyperparameter*, dan parameter yang dilakukan *tuning* adalah *optimizer*, *batch_size* dan juga *epochs*. Dalam proses *tuning hyperparameter* ini menggunakan library *GridSearchCV*. *Optimizer* yang akan diuji adalah *adam* dan *rmsprop*, *batch_size* 25, 32, 64 dan *epochs* 10, 15, 20.

Parameter terbaik yang didapatkan setelah *tuning hyperparameter* adalah *epochs*: 15, *optimizer*: *rmsprop*, dan *batch_size*: 64. Dengan rincian hasil pada Tabel-1. Visualisasi hasil *tuning*

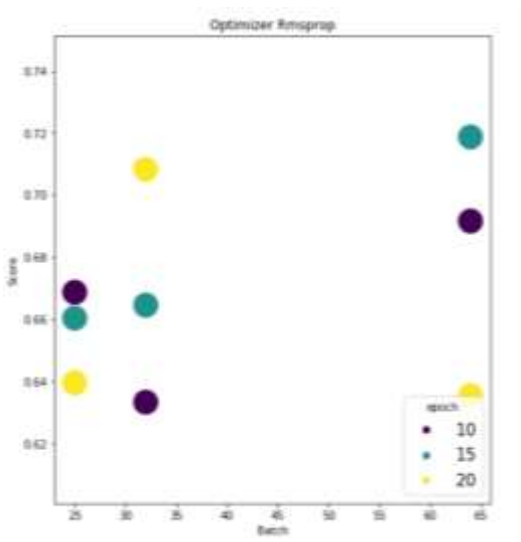
hyperparameter dapat dilihat pada Gambar-7 dan Gambar-8.

Tabel-1. Hasil *Tuning Hyperparameter*

Optimizer	Epoch	Batch	Akurasi		
Adam	10	25	0,631250		
		32	0,670833		
		64	0,675000		
		15	25	0,668750	
			32	0,654167	
			64	0,685417	
	20	25	0,664583		
		32	0,656250		
		64	0,687500		
		Rmsprop	10	25	0,668750
				32	0,633333
				64	0,691667
15	25		0,660417		
	32		0,664583		
	64		0,718750		
20	25	0,639583			
	32	0,708333			
	64	0,635417			

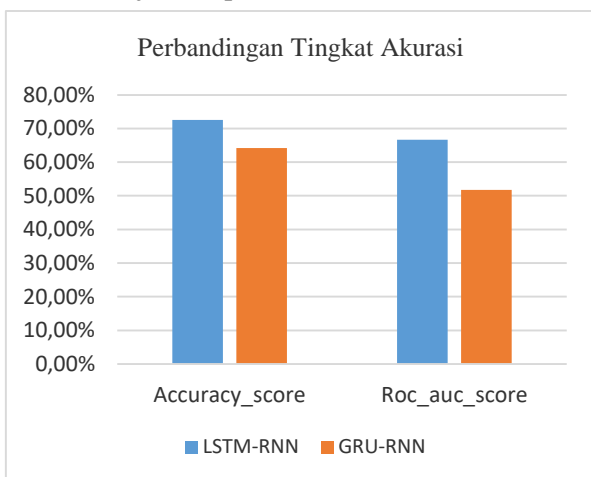


Gambar-7. Visualisasi Hasil *Tuning Hyperparameter* dengan *Optimizer Adam*

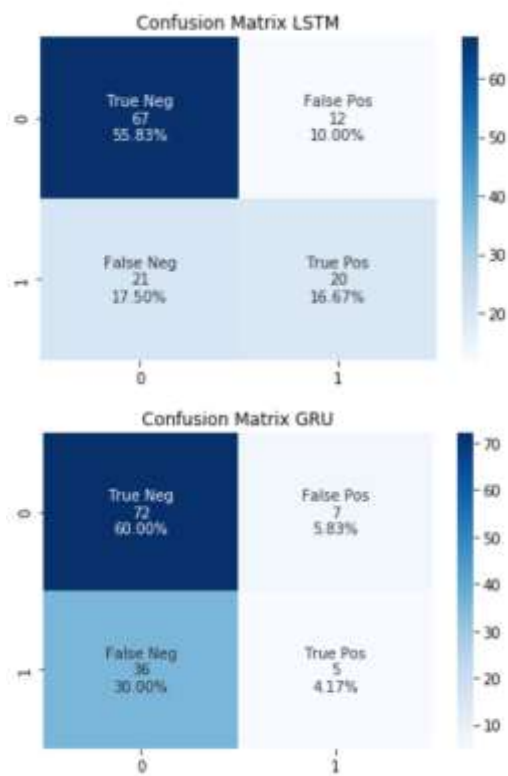


Gambar-8. Visualisasi Hasil *Tuning Hyperparameter* dengan *Optimizer Rmsprop*

Dilakukan evaluasi model sebelumnya dengan menggunakan parameter yang telah didapatkan dari proses *tuning*. Data di-*feed* kembali ke model yang telah dievaluasi dan didapat *score* 0,725 untuk *LSTM-RNN* dan 0,642 untuk *GRU-RNN* menggunakan library *accuracy_score*. Dan *score* 0,67 untuk *LSTM-RNN* dan 0,52 untuk *GRU-RNN* menggunakan *roc_auc_score*, seperti yang ditampilkan pada Gambar-9. Hasil dalam *confusion matrix* ditunjukkan pada Gambar-10.



Gambar-9. Grafik perbandingan tingkat akurasi LSTM-RNN dan GRU-RNN



Gambar-10. Confusion Matrix

Dari *confusion matrix* tersebut dapat dilihat bahwa ada 33 berita yang salah diprediksi oleh model LSTM-RNN dan 43 berita salah prediksi oleh model GRU-RNN dari total 120 *test data*. Lalu untuk nilai *precision*, F1 dan *recall* menggunakan library *classification_report* dengan hasil LSTM pada Tabel-2 dan hasil GRU pada Tabel-3.

Tabel-2. *Classification Report LSTM*

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
0	0,76	0,85	0,8	79
1	0,62	0,49	0,55	41
<i>accuracy</i>			0,73	120
<i>macro avg</i>	0,69	0,67	0,68	120
<i>weighted avg</i>	0,71	0,72	0,72	120

Tabel-3. *Classification Report GRU*

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
0	0,67	0,91	0,77	79
1	0,42	0,12	0,19	41
<i>accuracy</i>			0,64	120
<i>macro avg</i>	0,54	0,52	0,48	120
<i>weighted avg</i>	0,58	0,64	0,57	120

V. PENUTUP

Kesimpulan yang dapat diambil adalah:

1. Pada penelitian ini telah dilakukan klasifikasi berita palsu berbahasa Indonesia dengan membandingkan dua metode *Recurrent Neural Network* (RNN) yaitu *Long Short-Term Memories* (LSTM) dan *Gated Recurrent Unit* (GRU). Untuk mendapatkan kinerja optimal dalam klasifikasi berita palsu digunakan nilai parameter *epochs*: 15, fungsi *optimizer gradient descent* yaitu *rmsprop*, dan *batch_size*: 64 menggunakan metode LSTM.
2. Nilai akurasi yang dihasilkan oleh model cukup baik yaitu 73% menggunakan metode LSTM dan 64% menggunakan metode GRU. Namun nilai *recall* hanya untuk dapat mengklasifikasikan berita valid sebesar 85%, sedangkan untuk berita palsu sebesar 49%. Hal ini dikarenakan pada beberapa berita valid yang mengandung kata "hoax", "blokir", atau frasa yang berkaitan dengan opini seperti "cukupkah yakin masyarakat", "netizen resah informasi" diidentifikasi sebagai berita palsu. Sehingga dapat dilakukan proses yang dapat menangani permasalahan tersebut pada penelitian selanjutnya untuk meningkatkan nilai *recall*.

DAFTAR PUSTAKA

- [1] Pratiwi IYR, Asmara RA, Rahutomo F. *Study of Hoax News Detection Using Naive Bayes Classifier in Indonesian Language*. International Conference on Information & Communication Technology and System (ICTS). 2017;73-78.
 - [2] Al-ash HS, Putri MF, Mursanto P, Bustamam A. *Ensemble Learning Approach on Indonesian Fake News Classification*. 3rd International Conference on Informatics and Computational Sciences (ICICoS). 2019;2-7.
 - [3] Prasetijo AB, Isnanto RR, Eridani D, Alvin Y, Soetrisno A, Arfan M. *Hoax Detection System on Indonesian News Sites Based on Text Classification using SVM and SGD*. 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). 2017;45-49.
 - [4] Rusli A, ChristianYoung J, Iswari NMS. *Identifying Fake News in Indonesian via Supervised Binary Text Classification*. IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT). 2020;86-90.
 - [5] Trisna T, Putri A, Sitepu IY, Sihombing M, Indonesia UP, Petisah M. *Analysis and Detection of Hoax Contents in Indonesian News Based on Machine Learning*. *Journal of Informatic Pelita Nusantara*. 2019;4(1): 19-26.
 - [6] Apriliyanto A, Kusumaningrum R. *Hoax Detection in Indonesia Language using Long Short-Term Memory Model*. *Sinergi*. 2020;24(3):189-96.
 - [7] Bahad P, Saxena P, Kamal R. *Fake News Detection using Bi-directional LSTM-Recurrent Neural Network*. *Procedia Computer Science*. 2019;165:74-82.
 - [8] Suyanto S. *Synonyms-Based Augmentation to Improve Fake News Detection using Bidirectional LSTM*. 8th International Conference on Information and Communication Technology (ICoICT). 2020;8-12.
 - [9] Aditya Yanuar R. *Recurrent Neural Network (RNN)*. Universitas Gadjah Mada [Internet]. 9 Desember 2018; Tersedia pada: <https://machinelearning.mipa.ugm.ac.id>.
 - [10] Miranda ND, Novamizanti L, Rizal S. *Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50*. *Jurnal Teknik Informatika (Jutif)*. 2020; 1(2): 61-68.
 - [11] Hanifa A, Akbar S. *Detection of unstable approaches in flight track with recurrent neural network*. International Conference on Information and Communications Technology (ICoIACT). 2018;735-740.
 - [12] Saxena A, Sukumar TR, Nadu T, Nadu T. *Predicting bitcoin price using LSTM and Compare its predictability with Arima model*. *International Journal of Pure and Applied Mathematics*. 2018;119(17):2591-2600.
 - [13] Ribeiro AH, Schön TB. *Beyond exploding and vanishing gradients : analysing RNN training using attractors and smoothness*. International Conference on Artificial Intelligence and Statistics. 2020;2370-2380.
 - [14] Chung H, Shin K. *Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction*. *Sustainability*. 2018;10(10):3765.
 - [15] Chung J, Gulcehre C, Cho K, Bengio Y. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. *arXiv preprint. arXiv:1412.3555*. 2014.
-