Implementasi Long Short-Term Memory Untuk Identifikasi Berita Hoax Berbahasa Inggris Pada Media Sosial

(Implementation of Short-Term Short-Term Memory to Identify Hoax News in English on Social Media)

Jasman Pardede*, Raka Gemi Ibrahim
Jurusan Teknik Informatika, Institut Teknologi Nasional
Jl. PH. H. Mustofa No.23 dan Kota Bandung
Email: jasman@itenas.ac.id, raka.pancid@gmail.com

Penulis korespondensi

Abstract **Hoax or fake news spreads very fast on social media. The news can influence readers and poisoning their mind. It is important to identify the hoax news broadcasted on social media. Some methods to predict hoax are to use Support Vector Classifier, Logistic Regression, and MultinomialNaiveBayes. In this study, we applied Long Short-Term Memory to identify the hoaxes. System performance was measured based on the precision, recall, accuracy, and F-Measure scores. The experimental results, conducted on the hoaxes data obtained the average value of precision, recall, accuracy, and F-Measure of 0.94, 0.96, 0.95, and 0.95 respectively. The results shows that the proposed Long Short-Term Memory has better performance compared to the latest method.**

Key words**: Long Short-Term Memory, Hoax, Recurrent Neural Network.**

I. PENDAHULUAN

Dalam Kamus Besar Bahasa Indonesia (KBBI) kata hoax diserap menjadi hoaks yang diartikan sebagai informasi bohong. Kata hoax sendiri baru mulai digunakan sekitar tahun 1808 [1]. Kata tersebut dipercaya berasal dari kata *hocus* yang berarti 'untuk mengelabui'. Kata *hocus* sendiri merupakan singkatan dari *hocus pocus*, sejenis mantra yang kerap digunakan dalam pertunjukan sulap. Penyebaran informasi hoax bertujuan sebagai bahan lelucon, iseng, dan biasanya untuk menjatuhkan pesaing (*black campaign*). Dampak yang dihasilkan oleh hoax merupakan dampak yang tidak langsung disadari oleh pembaca, tetapi dapat menyerang pemikiran dan mempengaruhi cara berpikir pembaca jika tidak berhati-hati. Menurut Alessandro Bondielli [2] istilah hoax biasanya disebut sebagai "virus pikiran", hal ini dikarenakan kemampuannya untuk mereplikasi diri, mengadaptasi, memutasi, dan bertahan di dalam pikiran manusia.

Saat ini media sosial merupakan media komunikasi yang efektif dan efisien. Media sosial digunakan sebagai jembatan untuk membantu proses peralihan masyarakat yang tradisional ke masyarakat yang modern, khususnya untuk mentransfer informasi pembangunan yang dilaksanakan pemerintah kepada masyarakatnya [6]. Media sosial telah menjadi alat penerbitan penting bagi jurnalis dan metode konsumsi utama bagi masyarakat yang mencari berita terbaru.

Hoax dapat memberi pengaruh buruk pada seseorang melalui sebuah tulisan, dapat mempengaruhi pikiran waras seseorang, sementara itu, gambar dapat memunculkan rasa takut dan terancam. Menurut hasil riset Lembaga Ilmu Pengetahuan Indonesia (LIPI), masyarakat yang fanatik lebih mudah terkena hoax. Jika dibiarkan berita hoax dapat sangat persuasif, sehingga diperlukan strategi untuk mengidentifikasi berita hoax yang disebarkan di media sosial.

Penelitian [7] melakukan identifikasi berita hoax berbahasa Inggris pada media sosial. Mereka menggunakan dataset *fake_or_real_news* di proses menggunakan algoritma *machine learning* yaitu menggunakan metode *Logistic Regression* [7], kemudian menggunakan metode *MultinomialNaiveBayes* [7], dan yang terakhir menggunakan metode *Support Vector Classifier* [7]. Penelitian ini menggunakan metode LSTM untuk meng identifikasi berita hoax berbahasa Inggris pada dataset yang digunakan pada penelitian terdahulu yaitu dataset *fake_or_real_news*.

II. TINJAUAN PUSTAKA

A. Hoax

Pada hoax yang mengacu pada kamus jurnalistik yang dibuat [12], terdapat istilah *libel* yang berarti berita bohong yang berisikan tentang penghinaan, penistaan, pencemaran nama baik, hasutan, dan lain sebagainya di mana berita tersebut dapat merugikan orang lain baik yang dituangkan dalam tulisan dan secara lisan [5]. Tujuan hoax merupakan upaya untuk menipu pembaca untuk mempercayai sesuatu, padahal pembuat berita palsu itu tahu bahwa berita tersebut adalah palsu [2]. Berita hoax adalah sebuah pemberitaan yang terlihat seperti berita faktual, namun ternyata berisi kebohongan dan fitnah. Biasanya berita hoax sengaja dibuat untuk menyebarkan propaganda atau pesan kebencian atas seseorang atau instansi tertentu [1].

Contoh pemberitaan palsu yang paling umum adalah mengklaim suatu barang atau kejadian dengan suatu sebutan yang berbeda dengan barang/kejadian sejatinya. Menurut Dedi Rianto [4] hoax dikelompokan menjadi 3 jenis sebagai berikut:

1) *Fake News* atau berita bohong adalah salah satu jenis hoax. Berita bohong ini bertujuan untuk memalsukan kebenaran dalam suatu berita. Penulis berita bohong biasanya menambahkan hal-hal yang tidak benar.
2) *Clickbait* adalah jenis hoax yang merupakan suatu tautan berupa jebakan. Biasanya tautan tersebut diletakan secara strategis dalam suatu situs agar menarik orang untuk masuk ke dalam tautan tersebut. Tautan tersebut berupa fakta namun judulnya biasanya dilebih-lebihkan.
3) *Satire* adalah sebuah tulisan atau berita yang menggunakan humor, ironi, dan hal tersebut dibesar-besarkan untuk mengomentari suatu kejadian yang sedang hangat dibicarakan.

B. Word Embeddings

Stanford University menggunakan *GloVe* sebagai representasi kata untuk menghasilkan *word embeddings*[8]. Variasi dimensi yang dimulai dari 50 hingga 300 dimensi pada GloVe memudahkannya untuk menyesuaikan dengan dataset. Pada penelitian ini, *GloVe* menggunakan Korpus Wikipedia untuk membangun *vocabulary* di mana setiap kata pada *vocabulary* menghasilkan vektor yang berukuran ratusan dimensi. Model Glove ditetapkan berdasarkan:

$$w_i^t + \vec{w}_k + b_i + \vec{b}_k = log(Xi_k) \qquad (1)$$

di mana *w* adalah vektor kata, $\vec{w}$ adalah vektor konteks kata, b_i dan b_k adalah bias skalar untuk kata utama dan konteks kata. *X* adalah matriks kemunnculan di mana Xi_k mempresentasikan jumlah berapa kali kata k muncul di konteks kata i. $f(Xi_k)$ fungsi bobot. Perhitungan Xi_k didapatkan dengan cara menghitung statistik kemunculan kata dalam bentuk matrik kemunculan *x*. Setiap elemen matriks Xi_k mewakili seberapa sering sebuah kata muncul dalam konteks kata *j*, di mana konteks kata merupakan kumpulan kata yang terdiri atas kata-kata yang berada sebelum dan sesudah kata i sepanjang *windows size* yang diberikan. Pembobotan kata untuk setiap kata dalam konteks kata menggunakan 1 *distance* , distance di sini dihitung berdasarkan panjang konteks kata posisi kata tersebut. Nilai $f(Xi_k)$ dihitung menggunakan Persamaan (2).

$$f(Xi_k) = \begin{cases} \left(\frac{Xi_k}{xmax}\right)^a ; if\, Xi_k < xmax \\ 1; lainnya \end{cases} \qquad (2)$$

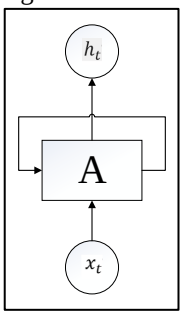
Model *GloVe* pada persamaan (2) memperkenalkan fungsi pembobotan ke dalam fungsi *cost* yang memberikan model seperti pada Persamaan (3).

$$J = \sum_{i,k=1}^{V} f(Xi_k)(w_i^t \vec{w}_J + b_i + b_k - logXi_k) \qquad (3)$$

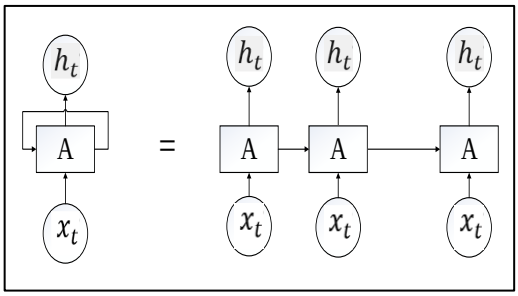
Pada model *GloVe*, parameter yang digunakan antara lain *xmax*, *alpha*, dan iterasi. Nilai paramater yang digunakan pada tugas akhir ini adalah *xmax* 100 dan *alpha* ¾, merujuk pada paper *GloVe*: *Global Vectors for Word Representation* yang menggunakan dataset wikipedia bahasa Inggris dengan nilai *xmax* 100 dan memberikan performa yang baik walaupun dengan dimensi vektor yang kecil [8]. Untuk parameter iterasi, *Glove* dapat menggunakan nilai iterasi yang beragam dan semakin besar nilai iterasi akan menghasilkan performa yang lebih baik [8]. Penelitian ini menggunakan menggunakan 50 iterasi.

C. Recurrent Nerual Network

Recurrent Neural Network atau biasa disingkat RNN adalah jenis jaringan syaraf tiruan untuk memproses data sekeunsial seperti pengenalan ucapan, dan pemodelan bahasa [10]. Pemrosesan RNN dilakukan secara berulang. Gambar 1 merupakan proses perulangan *RNN*. Gambar 2 merupakan salinan jaringan RNN.



Gambar 1. Proses Perulangan *Recurrent Neural Network* [10]



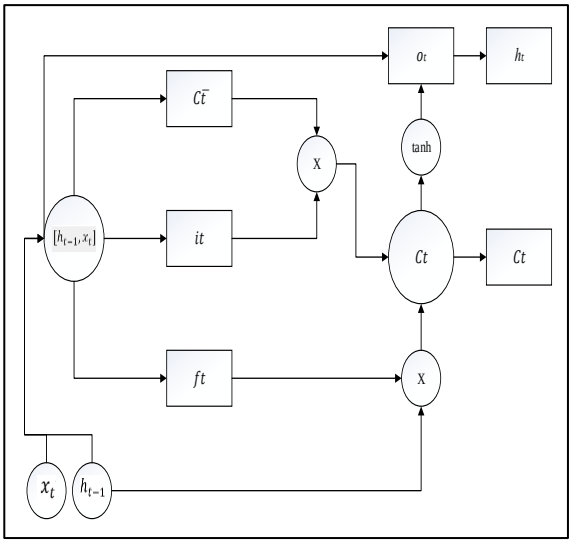
Gambar 2. Salinan Jaringan Pada *Recurrent Neural Network* [10]

Pada Gambar 2, x_t sebagai input, h_t sebagai output, dan terdapat alur perulangan yang memungkinkan informasi dilewatkan dari satu langkah jaringan ke langkah berikutnya [10]. *Recurrent eural network* juga memiliki banyak salinan yang sama, masing-masing menyampaikan pesan kepada penerus. Dalam satu *recurrent neural network* terdapat banyak salinan yang sama. x_t sebagai input, h_t sebagai output dan terdapat alur perulangan yang memungkinkan informasi dilewatkan

dari satu langkah jaringan ke langkah berikutnya. Satu masalah pada arsitektur RNN adalah masalah ketergantungan jangka panjang. Masalah tersebut diatasi menggunakan variasi RNN yaitu *Long Short Term Memory* (LSTM) [9].

D. Long Short-Term Memory

Long Short Term Memory adalah salah satu variasi *RNN* yang dibuat untuk menghindari masalah ketergantungan jangka panjang pada RNN [9]. LSTM dapat mengingat informasi jangka panjang [10]. Sama seperti *RNN*, LSTM juga terdiri dari modul pemrosesan berulang. Gambar 3 menunjukkan arsitektur LSTM.



Gambar 3. Arsitektur LSTM.

Ide dari LSTM adalah dibuatnya jalur yang menghubungkan konteks lama c_{t-1} ke konteks baru (c_t) yang disebut juga *cell state*, *memory cell* atau jalur memori [10]. Dengan adanya jalur tersebut, suatu nilai pada konteks yang lama akan dengan mudah dihubungkan ke konteks yang baru jika diperlukan dengan sedikit sekali modifikasi. LSTM memiliki kemampuan untuk menghapus atau menambahkan informasi ke keadaan sel, dan diatur dengan cermat oleh fungsi *sigmoid*. Langkah-langkah pada LSTM memiliki 4 gerbang layer yaitu *forget gate* (4), *input gate* (5) (6), *cell gate* (7), dan *output gate* (8) (9).

1) Forget Gate

Forget gate adalah *gate* yang memutuskan apakah suatu informasi harus dibuang atau tidak dari pemrosesan. *Gate* ini bernilai 0-1; jika 1 maka informasi disimpan jika 0 maka informasi dihapus.

$$f_t = \sigma(w_f.[h_{t-1}, x_t] + b_f) \quad (4)$$

Keterangan :
f_t = *forget gate*
σ = fungsi *sigmoid*
W_f = *weight* untuk *forget gate*
h_{t-1} = output sebelum orde ke-*t*
x_t = input pada orde ke-*t*
b_f = bias pada *forget gate*

2) Input Gate

Input Gate adalah *gate* yang memutuskan untuk menentukan sebuah masukan akan ditambahkan ke dalam memori *cell gate*.

$$i_t = \sigma(w_i.[h_{t-1}, x_t] + b_i) \quad (5)$$

Keterangan :
i_t = *input gate*
σ = fungsi *sigmoid*
Wi = *weight* untuk *input gate*
h_{t-1}= output sebelum orde ke-*t*
x_t = input pada orde ke-*t*
b_i = bias pada *input gate*

$$\tilde{c}_t = \tanh(w_c.[h_{t-1}, x_t] + b_c) \quad (6)$$

Keterangan :
$\tilde{C}_t$ = kandidat konteks baru yang akan ditambahkan ke *cell gate*
$tanh$ = fungsi *tanh*
Wc = *weight* untuk *cell state*
h_{t-1} = output sebelum orde ke-*t*
x_t = input pada orde ke-*t*
b_C = bias untuk *cell state*

3) Cell Gate / Memory State

Cell gate adalah gate yang berfungsi sebagai memori untuk sebuah layer. *Cell gate* digunakan untuk mengingat informasi jangka panjang.

$$c_t = f_t * c_{t-1} + i_t.\tilde{c}_t \quad (7)$$

Keterangan :
C_t = *cell state*
ft = *forget gate*
C_{t-1}= *cell state* sebelum orde ke-*t*
it = *input gate*
$\tilde{C}_t$ = kandidat konteks baru yang dapat ditambahkan ke *cell state*

4) Output Gate

Output gate adalah *gate* yang berfungsi memutuskan apa yang akan dihasilkan berdasarkan *input* dan *cell gate*.

$$o_t = \sigma(w_o.[h_{t-1}, x_t] + b_o) \quad (8)$$

Keterangan:
o_t = *output gate*
σ = fungsi *sigmoid*
Wo = nilai *weight* untuk *output gate*

h_{t-1} = nilai output sebelum orde ke-*t*
x_t = nilai input pada orde ke-*t*
b_o = nilai bias pada *output gate*

$$h_t = o_t * \tanh(c_t) \quad (9)$$

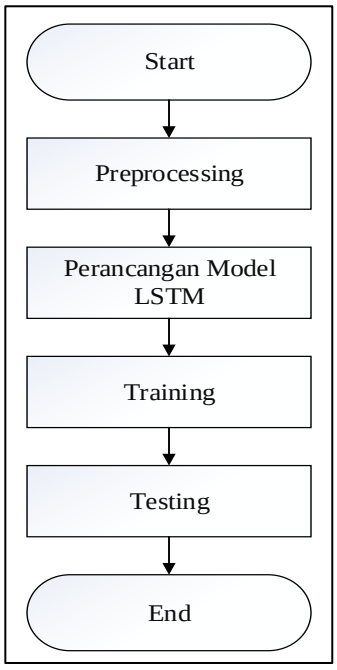
Keterangan:
h_t = output orde t
o_t = *output gate*
$tanh$ = fungsi *tanh*
C_t = *cell state*

III. METODE PENELITIAN

Secara umum, terdapat beberapa tahapan yang dilakukan untuk membangun model LSTM yaitu tahap preprocessing, perancangan model LSTM, training, dan testing. Tahapan pembentukan model LSTM ditunjukan pada Gambar 4.

A. Dataset

Pada penelitian ini dataset yang dipakai adalah *fake_or_real_news.csv* yang diperoleh dari situs *kaggle.com*. Dataset ini berisi *column unnamed*, *title*, *text*, dan *label*. Kemudian dataset yang digunakan akan dibagi menjadi rasio 80-10-10, di mana 80% digunakan sebagai data training, 10% sebagai data validation, dan 10% sebagai data test. Dataset yang diproses berbentuk kolom *text* dan *label*. Kolom label adalah "*fake*" dan "*real*". Pelabelan pada dataset yang dilakukan secara *crowdsourcing*. "0" merepresentasikan hoax atau "*fake*" dan "1" merepresesntasikan kebenaran atau "*real*". Gambar 5 merupakan tampilan dataset dalam format csv.



Gambar 4. Tahapan Penelitian

Unnamed: 0	title	text	label
8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

Gambar 5. Dataset yang didapat dari *kaggle.com*

B. Preprocessing

Sebelum membuat perancangan model LSTM, dilakukan tahap *preprocessing* yaitu *case folding*, *punctuation removal*, *stopword removal*, dan *tokenization*. Tahap *preprocessing* dilakukan dalam beberapa tahap sebagai berikut:

1) Case Folding

Penerapan *case folding* bertujuan untuk menyamakan seluruh jenis karakter yang terdapat dalam teks berita sehingga memudahkan proses penghapusan karakter atau kata-kata tertentu yang tidak diinginkan. Pada penelitian ini, semua huruf dalam dokumen dirubah menjadi huruf kecil. Hanya huruf 'a' hingga 'z' yang diterima.

2) Punctuation Removal

Pada tahap ini, tanda baca yang dimaksud adalah (? ! , / = + - \ > < ; " ( ) {} [ ] . = |) dan lainnya. Penghapusan ini dilakukan karena tanda baca diabaikan selama proses training sehingga penghapusan tanda baca akan menyederhanakan proses training.

3) Stopword Removal

Pada tahap ini, setelah menyamakan huruf menjadi huruf kecil dan menghilangkan tanda baca kemudian dilakukan filtering yaitu menghilangkan atau membuang kata yang kurang penting maknanya sehingga mesin hanya memproses kata yang bermakna.

TABEL I. CONTOH DATASET YANG TELAH DI-*PREPROCESSING*.

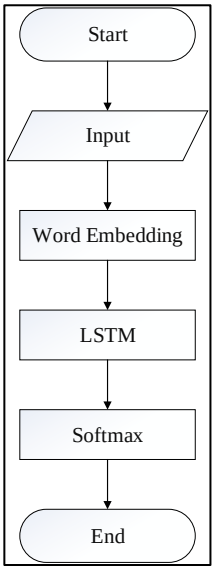
No	***Text***	***Label***
1	['google', 'pinterest', 'digg', 'linkedin', 'reddit', 'stumbleupon', 'print', 'delicious', 'pocket', 'tumblr', 'two', 'fundamental', 'truths', 'world', 'paul', 'ryan', 'desperately', 'wants', 'president', 'paul', 'ryan', 'never', 'president', 'today', 'proved', 'particularly', 'staggering', 'example', 'political', 'cowardice', 'paul', 'ryan', 'rerereversed', 'course', 'announced', 'back', 'trump', 'train', 'about', 'weeks', 'ago', 'previously', 'declared', 'would',	0

No	*Text*	*Label*
2	'supporting', 'defending', 'trump', 'tape', 'made', 'public', 'trump', 'bragged', 'assaulting', 'women', 'suddenly', 'ryan', 'appearing', 'protrump', 'rally', 'boldly', 'declaring', 'already', 'sent', 'vote', 'make', 'president', 'united', 'states'] ['us', 'secretary', 'state', 'john', 'kerry', 'said', 'monday', 'stop', 'paris', 'later', 'week', 'amid', 'criticism', 'top', 'american', 'officials', 'attended', 'sundays', 'unity', 'march', 'terrorism', 'kerry', 'said', 'expects', 'arrive', 'paris', 'thursday', 'evening', 'heads', 'home', 'week', 'abroad', 'said', 'fly', 'france', 'conclusion', 'series', 'meetings', 'scheduled', 'thursday', 'sofia', 'bulgaria', 'plans', 'meet', 'next', 'day', 'foreign', 'minister', 'laurent', 'fabius', 'president', 'francois', 'hollande', 'return', 'washington', 'visit', 'kerry', 'family', 'childhood', 'ties', 'country', 'speaks', 'fluent', 'french', 'could', 'address', 'criticism', 'united', 'states', 'snubbed', 'france', 'darkest', 'hour', 'many', 'years', 'french', 'press', 'monday', 'filled', 'questions', 'neither', 'president', 'obama',]	1

4) Tokenization

Tokenisasi berfungsi memecah teks manjadi kata atau melakukan pengubahan seluruh kalimat yang ada pada data menjadi sebuah kata atau *token* agar mesin mudah memproses data tersebut.

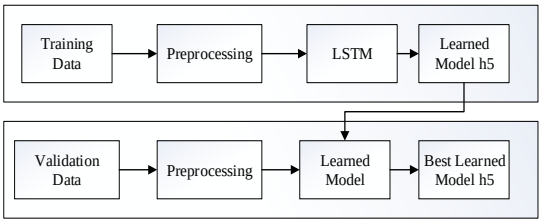
Tabel I merupakan contoh dataset yang telah melalui tahap preprocessing.



Gambar 5. Perancangan Model LSTM

C. Training

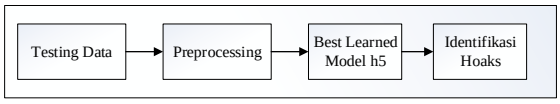
Proses training adalah pembuatan model dari LSTM. Proses tersebut diawali dengan mengambil data berita hoax dari *kaggle.com*; data tersebut sudah dilabeli *fake* (hoax) dan *real* (non hoax). Kemudian dilakukan *pre processing* pada data tersebut dengan menggunakn *case folding*, *puncutaion removal*, *stopwords removal*, dan *tokenization*. Setelah itu barulah masuk pada tahap awal model LSTM, yaitu *word embedding* agar data yang telah diproses menghasilkan tensor 3D berupa vektor yang akan diproses oleh LSTM. Setelah menghasilkan bentuk tensor 3D, baru diproses oleh LSTM dengan empat gate di dalamnya yaitu *forget gate*, *input gate*, *cell gate*, dan *output gate*. Setelah selesai, hasil prosesnya disimpan menjadi model dengan ekstensi .h5. Tentu untuk menghasilkan model yang baik harus mengukur kinerja dari model yang telah dilatih dengan data validation. Gambar 6 merupakan tahapan training tersebut.



Gambar 6. Tahapan Training

D. Testing

Pada tahap ini, setelah proses training, untuk mengetahui kinerja model yang dibuat, dilakukan testing menggunakan model terbaik dengan data uji 10% dari keseluruhan dataset. Tahapan testing ditunjukkan oleh Gambar 7.



Gambar 7. Tahapan Testing

E. Evaluasi

Evaluasi dilakukan menggunakan *precision*, *recall*, *accuracy*, dan *f−measure* untuk mengukur kinerja model LSTM menggunakan *confusion-matrix*. *Confusion-matrix* yang digunakan dapat dilihat pada Gambar 22. *precision*, *recall*, *accuracy*, dan *f-measure* [11] dihitung menggunakan Persamaan (10) hingga (13).

$$precission = \frac{TP}{TP+FP} \quad (10)$$

$$recall = \frac{TP}{TP+FN} \quad (11)$$

$$accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (12)$$

$$F\ Measure = 2\ x\ \frac{(Precision\ x\ Recall)}{(Precision+Recall)} \quad (13)$$

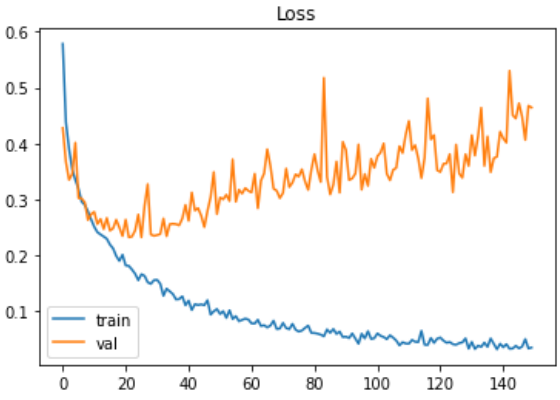
E-ISSN:2541-0806
P-ISSN:2540-8895

Keterangan =
TP=Dikategorikan hoax; sebenarnya memang hoax
TN=Dikategorikan *real* atau non-hoax; sebenarnya memang *real* atau non-hoax
FP=Dikategorikan hoax namun sebenarnya *real* atau non-hoax
FN=Dikategorikan *real* atau non-hoax namun sebenarnya hoax.

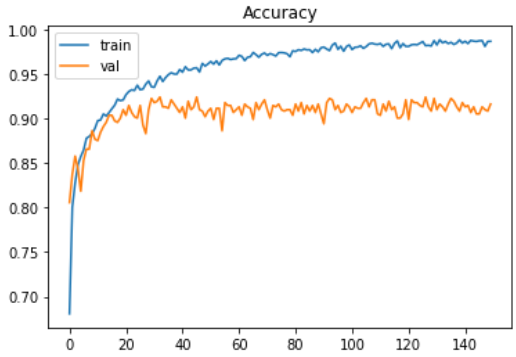
IV. HASIL DAN PEMBAHASAN

Penelitian ini melakukan proses training pada mesin dengan spesifikasi *processor* Intel(R) Xeon(R) CPU @ 2.20GHz, RAM 13GB, dan *storage* 33GB yang disediakan oleh Google Collaboratory. Proses testing menggunakan *processor* Intel Core i5-7200U @2.5GHz, RAM 4GB, storage 1TB. Penelitian ini menggunakan Tensorflow dan Keras.

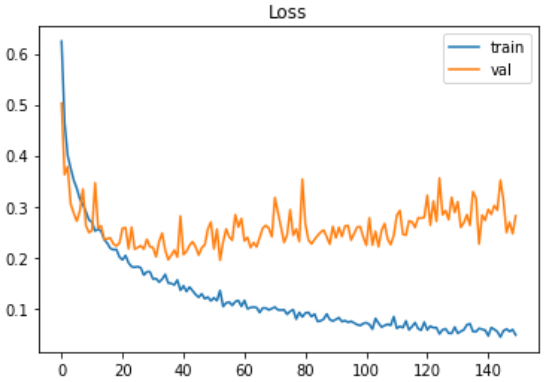
Penelitian ini bertujuan mengetahui nilai probabilitas tertinggi berdasarkan kategori hoax dan real (non-hoax) dengan mengimplementasikan metode LSTM. Untuk training data dibagi dengan rasio 80% data train, 10% data validation dan 10% data data test. Model dibangun menggunakan 150 *epoch*, 32 *batch size*, dan 50 layer LSTM. Pada penelitian ini, dilakukan eksperimen menggunakan *dropout* dengan nilai mulai dari 0.20, 0.25, 0.3, 0.35, 0.4, 0.45 dan 0.5 dengan *epoch* dan *batch size* yang sama. Pada Gambar 8 hingga 20, axis (*x*) adalah informasi *epoch* mulai dari 0 hingga 150. Kemudian axis (*y*) adalah informasi nilai untuk setiap *loss* dan *accuracy*. Garis yang berwarna biru merupakan alur grafik dari data latih dan garis berwarna oranye adalah alur grafik dari data validasi. Gambar 8, 10, 12, 14, 16, 18, dan 20 adalah grafik *loss* dengan nilai *dropout* 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, dan 0.5 sementara Gambar 9, 11, 13, 15, 17, 19, dan 21 adalah *accuracy* untuk masing-masing *dropout*.



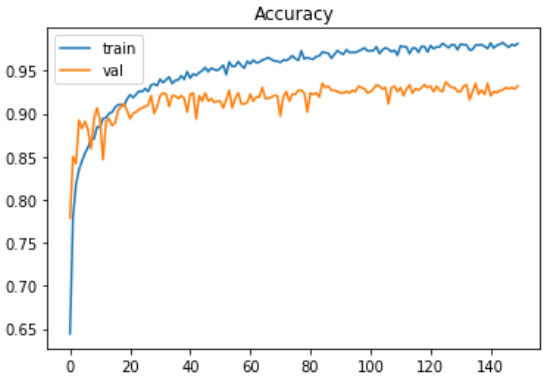
Gambar 8. Hasil *Loss* dengan Nilai *Dropout* 0,2.



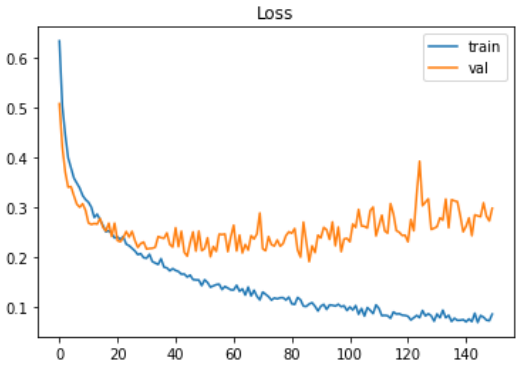
Gambar 9. Hasil *Accuracy* dengan Nilai *Dropout* 0.2



Gambar 10. Hasil *Loss* dengan Nilai *Dropout* 0.25

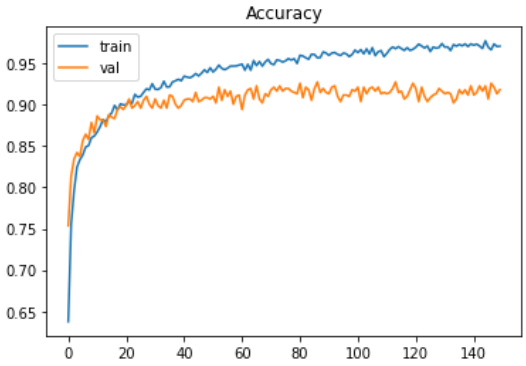


Gambar 11. Hasil *Accuracy* dengan Nilai *Dropout* 0,25.

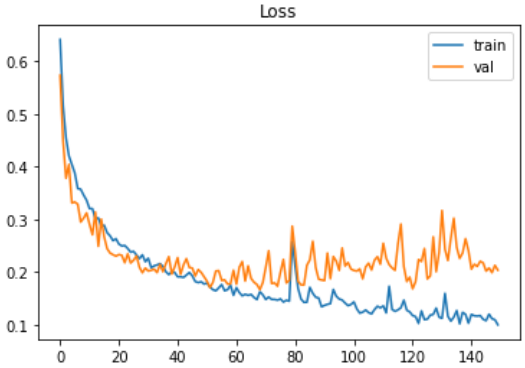


Gambar 12. Hasil *Loss* dengan Nilai *Dropout* 0,3.

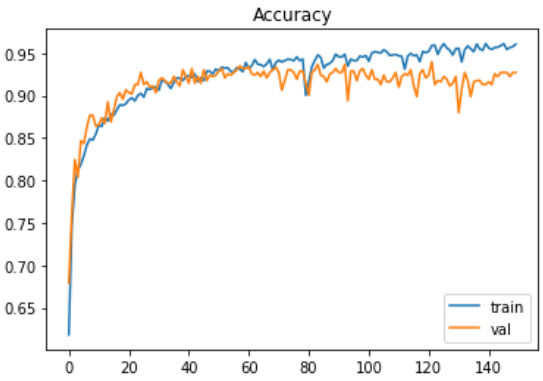
E-ISSN:2541-0806
P-ISSN:2540-8895



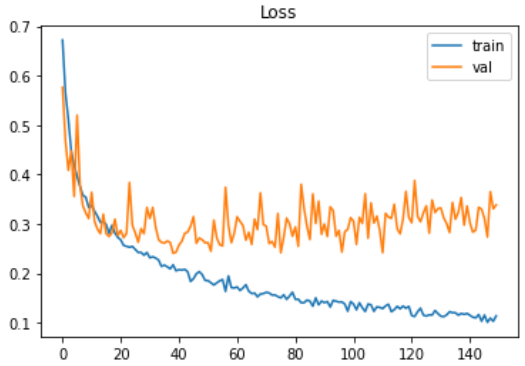
Gambar 13. Hasil *Accuracy* dengan Nilai *Dropout* 0,3.



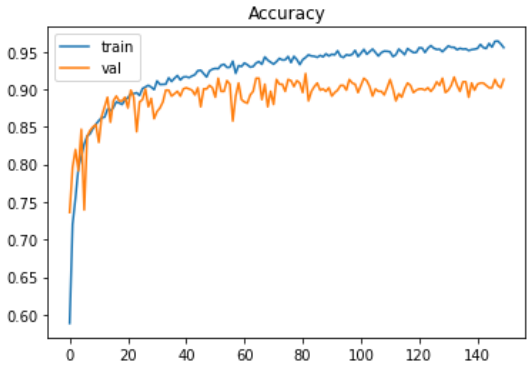
Gambar 14. Hasil *Loss* dengan Nilai *Dropout* 0,35.



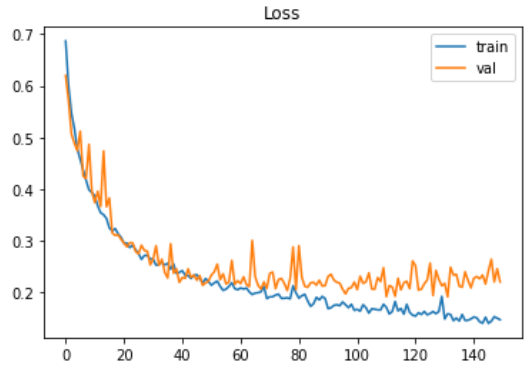
Gambar 15. Hasil *Accuracy* dengan Nilai *Dropout* 0,35.



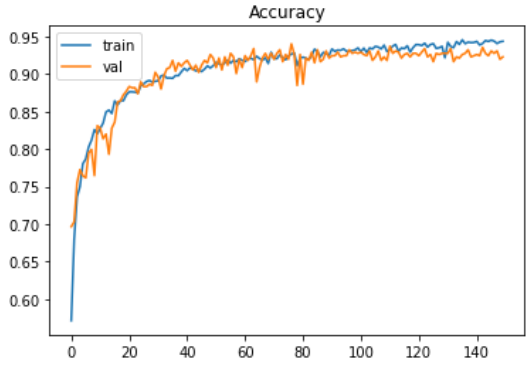
Gambar 16. Hasil *Loss* dengan Nilai *Dropout* 0,4.



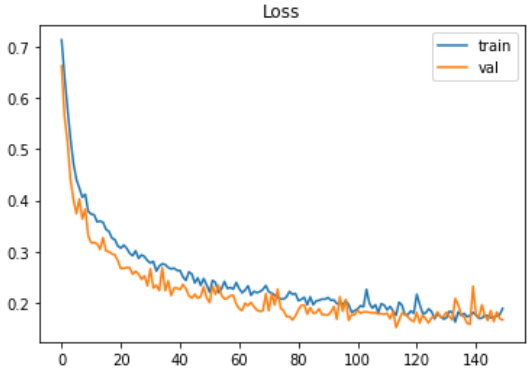
Gambar 17. Hasil *Accuracy* dengan Nilai *Dropout* 0,4.



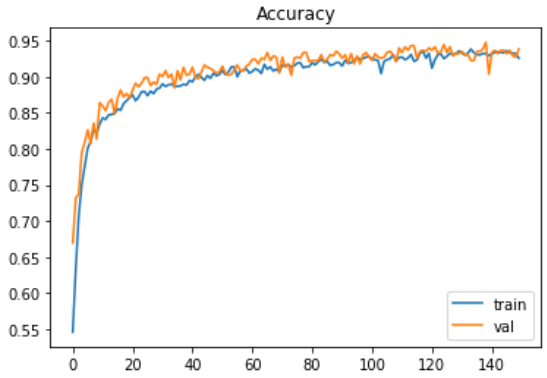
Gambar 18. Hasil *Loss* dengan Nilai *Dropout* 0,45.



Gambar 19. Hasil *Accuracy* dengan Nilai *Dropout* 0,45.

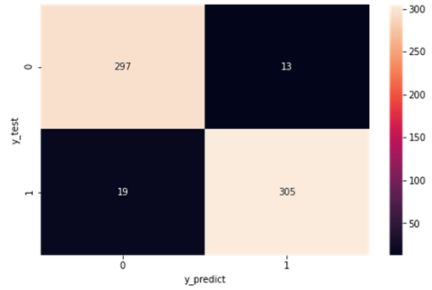


Gambar 20. Hasil *Loss* dengan Nilai *Dropout* 0,5.



Gambar 21. Hasil *Accuracy* dengan Nilai *Dropout* 0,5.

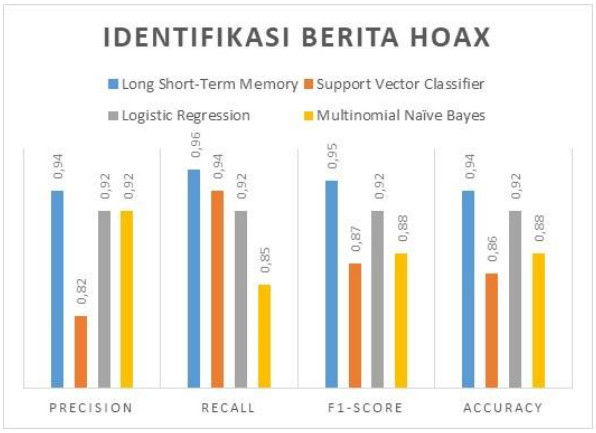
Pada Gambar 8 sampai dengan Gambar 21, terlihat perubahan *loss* dan *accuracy* saat proses training dengan nilai dropout 0.2 sampai dengan 0.5. Berdasarkan grafik-grafik tersebut, eksperimen yang menggunakan nilai *dropout* 0.5 menunjukan kinerja terbaik daripada yang menggunakan nilai *dropout* 0.2 sampai dengan 0.4. Kemudian setelah mendapatkan kinerja terbaik model LSTM dari proses training, model tersebut diuji dengan menggunakan data test yang berukuran 10%. Untuk mengukur kinerja model yang dihasilkan digunakan menggunakan *confusion-matrix* dengan parameter *precision*, *recall*, *accuracy* dan *f-measure*. Gambar 22 merupakan hasil *confusion-matrix*.



Gambar 22. Confusion-Matrix

Berdasarkan Gambar 22, seluruhnya terdapat 634 data test di mana 297 TP) data test hoax yang diprediksi sebagai hoax dan 19 data test yang sebeneranya real atau non-hoax namun diprediksi sebagai hoax (FP). Kemudian sebanyak 305 data test non hoax yang diprediksi real atau non-hoax (TN) dan 13 data test hoax yang dideteksi non-hoax (FN). Dengan demikian, diperoleh 297 TP, 19 FP, 13 FN, dan 305 TN sehingga nilai *precision*, *recall*, *accuracy*, dan *f-measure* adalah 0.94, 0.96, 0.94, dan 0,95.

Kemudian penelitian [7] melakukan pengujian pada model *support vector classifier*, model *Logistic Regression*, dan *MultinomialNaiveBayes* [7] untuk dataset yang sama (*fake_or_real_news*). Model *support vector classifier* mendapat *precision*, *recall*, *accuracy*, dan *f-measure* sebesar 0,82, 0,94, 0,87, dan 0,95. Pada model *Logistic Regression* dicapai *precision*, *recall*, *accuracy*, dan *f-measure* masing-masing 0,92, sementara pada model *MultinomialNaiveBayes* didapatkan *precision*, *recall*, *accuracy*, dan *f-measure* 0,92, 0,85, 0,88, dan 0,88. Gambar 23 merupakan perbandingan kinerja *precision*, *recall*, *accuracy*, dan *f-measure* untuk masing-masing model di mana LSTM unggul dari ketiga model lainnya.



Gambar 23. Hasil perbandingan kinerja *precision*, *recall*, *accuracy*, dan *f−measure* dari LSTM dengan metode lain.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Pada penelitian ini telah diimplementasikan metode LSTM untuk mengindentifikasi berita hoax berbahasa Inggris pada media sosial. Metode yang diusulkan dapat melakukan identifikasi berita hoax dengan nilai rata-rata *precision*, *recall*, *accuracy*, dan *f-measure* sebesar 0,94, 0,96, 0.94 dan 0.95. Hasil penelitian tersebut menunjukkan bahwa LSTM memberikan kinerja terbaik dibandingkan metode *Support Vector Classifier*, *Logistic Regression*, dan *MultinomialNaiveBayes*.

B. Saran

Penelitian selanjutnya diharapkan dapat meningkatkan kinerja untuk membangun *vocabulary* dengan memilih korpus yang lebih variatif, sehingga memiliki *vocabulary* yang lebih bervarisi.

DAFTAR PUSTAKA

[1] Aditiawarman, “Hoax Dan Hate Speech di Dunia Maya,” Lembaga Kajian Aset Budaya Indonesia, 2019.

[2] A. Bondielli, “A survey on Fake News and Rumour Detection Techniques”., 1892.

[3] Anshar, Pengaruh Hoax Bagi Kehidupan dan Bernegara, 2019.

[4] D. R. Rahadi, Perilaku Pengguna dan Informasi Hoax di Media Sosial, 2017.

[5] Y. R. Silitonga, Munawar, and I. N. Hapsari, Analisis dan Penerapan Data mining untuk Mendeteksi Berita Palsu (Fake News) pada Social Media dengan Memanfaatkan Modul Scikit Learn, 2019.

[6] A. Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata and Rob Procter, Detection and Resolution of Rumours in Social Media: A Survey, 2018.

[7] O. E. Taylor, P. S. Ezekiel, and J. Palimote, Application of Supervised Machine Learning Algorithms to Detect Online Fake News, 2020.

[8] J. Pennington, R. Socher, and Christopher D. Manning, GloVe: Global Vectors for Word Representation, Computer Science Department, Stanford University, Stanford, CA 94305, 2014.

[9] Sepp Hochreiter, and J. Schmidhuber, Long Short-Term Memory, 1997.

[10] Olah. C, Understanding LSTM Network, https=//colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

[11] J. Pardede, and M. G. Husada, "Comparison of SVM, GVSM, and LSI in Information Retrieval for Indonesian Text," vol. 78 (5–6), hal. 51–56, 2016.

[12] A. S. M. Romli, Kamus jurnalistik: Daftar Istilah penting Jurnalistik Cetak, Radio, dan Televisi, 2010.