

PERBANDINGAN ANALISIS SENTIMEN INSTAGRAM TENTANG PENANGANAN BANJIR DENGAN METODE SVM DAN RNN TIPE LSTM

Herry Wijaya (065116076)

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Pakuan

Alamat : herrywijaya065116076@unpak.ac.id

Abstrak

Instagram adalah salah satu media sosial yang paling populer di Indonesia. Pemerintah menggunakannya sebagai media penyampaian informasi terkait aspek umum ataupun permasalahan-permasalahan untuk masyarakat. Salah satu contohnya adalah memberikan informasi kegiatan yang sedang berlangsung atau akan direncanakan terkait penanganan banjir yang dilakukan di Jakarta. Dengan hal ini, maka masyarakat dapat mengetahui informasi penting dan juga memberikan komentar terkait kegiatan-kegiatan tersebut. Penelitian ini merupakan analisis sentimen yang membandingkan performa model algoritma SVM (*Support Vector Machine*) dan RNN tipe LSTM (*Recurrent Neural Network*) menggunakan dataset komentar instagram tentang penanganan banjir yang terjadi di Jakarta. Analisis sentimen ini bertipe klasifikasi biner dengan dua feature, yaitu data teks komentar sebagai input observasi dan polaritas sentimen positif dan negatif sebagai label kelas yang akan diprediksi. Dataset komentar instagram yang dilatih menggunakan kedua model algoritma akan dievaluasi agar diketahui seberapa baik model tersebut dalam memprediksi apakah suatu komentar berpolaritas sentimen positif atau negatif. Pengukuran evaluasi yang dihasilkan kedua model algoritma dibandingkan menggunakan *confusion matrix*. Hasil penelitian membuktikan bahwa model algoritma RNN tipe LSTM memiliki performa yang lebih baik daripada model algoritma SVM. Model LSTM memiliki akurasi sebesar 78,84%.

Kata kunci : Analisis Sentimen, *Long Short Term Memory* (LSTM), konfigurasi *hyperparameter*, *confusion matrix*, klasifikasi biner.

Abstract

Instagram is one of the most popular social media in Indonesia. Government uses it as information delivery media regarding general aspects or problems for public. One of the examples is to give activities' information which is going on or will be planned regarding flood management done in Jakarta. With this, public can find out important information and also give comments regarding those activities. This research is sentiment analysis which compares SVM (*Support Vector Machine*) and RNN-LSTM type (*Long Short Term Memory*) algorithm models' performance using instagram's comments dataset about flood management done in Jakarta. This sentiment analysis is a binary classification type with two features, which are comments text data as observation inputs and positive and negative sentiment polarity as class labels that will be predicted. Trained instagram comments dataset using two algorithm models will be evaluated so it can be known how good those models in predicting whether a comment has a positive or negative sentiment polarity. Evaluation measurement generated by two algorithm models will be compared using confusion matrix. Research result proves that RNN-LSTM type model algorithm has better performance than SVM algorithm model. LSTM model has 78,84% accuracy.

Kata kunci : *Sentiment Analysis*, *Long Short Term Memory* (LSTM), *hyperparameter configuration*, *confusion matrix*, *binary classification*.

1. Pendahuluan

Sejak awal 1 Januari 2020 sudah sering terjadi musibah banjir di ibukota Jakarta dan area metropolitan sekitarnya. Musibah ini merugikan harta benda dan merusak fasilitas umum milik masyarakat serta memakan korban jiwa yang tidak sedikit jumlahnya. Banjir yang terjadi disebabkan oleh curah hujan yang deras sehingga sungai Ciliwung dan Cisadane tidak mampu menjaganya melebihi kapasitas. Alasan lainnya adalah jumlah penduduk yang tidak terkontrol dan kurangnya pemahaman tentang perencanaan lahan.

Analisis Sentimen adalah proses otomatis yang menentukan apakah bagian suatu teks mengandung konten objektif atau berupa opini yang dapat dianalisis lebih lanjut dengan polaritas sentimen teks tersebut [5]. Analisis sentimen dapat digunakan untuk menganalisis sentimen masyarakat terhadap permasalahan banjir yang terjadi. Hal ini memerlukan data teks yang mewakili komentar masyarakat sebagai bahan analisis, sehingga sosial media merupakan salah satu channel yang dapat diambil untuk menggali data yang diperlukan dalam penelitian. Beberapa metode klasifikasi yang digunakan dalam analisis sentimen adalah RNN dan SVM. RNN merupakan bagian dari *Deep Learning* dimana nilai akurasi yang dihasilkan dapat lebih baik dibandingkan dengan jaringan syaraf tiruan sederhana karena bobot yang dihitung akan lebih akurat mendekati persamaan setiap kata. Jenis RNN yang digunakan adalah LSTM (*Long Short Term Memory*) untuk menutupi kekurangan pada RNN yang tidak dapat menyimpan memori untuk dipilah dan menambahkan mekanisme attention agar setiap kata dapat lebih fokus pada konteks [4]. Sedangkan SVM (*Support Vector Machine*) merupakan algoritma *machine learning* yang sudah sering digunakan dalam mendeteksi polaritas suatu teks [1]. Kelebihan SVM dalam *sentiment analysis* adalah memiliki performa yang baik jika jumlah sampel lebih banyak daripada jumlah *feature* dan dataset memiliki margin pemisahan yang jelas dalam setiap kelas.

Berdasarkan permasalahan dan penelitian terkait, maka dibuatlah perbandingan analisis sentimen instagram tentang penanganan banjir dengan menggunakan metode RNN dan SVM. Diharapkan penelitian ini dapat bermanfaat untuk mengetahui perbedaan antara *deep learning* dengan *machine learning* tradisional. Perbedaan ini meliputi arsitektur model yang lebih rumit dan adanya beragam *hyperparameter* unik dalam setiap layer jaringan syaraf tiruan. Hal ini memberikan kesempatan untuk beragam konfigurasi yang berbeda dalam pelatihan model sehingga diperoleh output yang berbeda, apakah lebih baik atau lebih buruk. Dengan demikian, *deep learning* memberikan kesempatan yang lebih luas untuk membangun model yang cocok dan optimal.

2. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah metodologi *Knowledge Discovery and Databases*. Metode ini berguna untuk memperoleh pengetahuan penting dari kumpulan data yang diproses dan memiliki lima tahap, yaitu tahap seleksi, tahap *preprocessing*, tahap transformasi, tahap *data mining*, dan tahap interpretasi dan evaluasi. Gambar 1 menunjukkan *flowchart* program analisis sentimen yang akan dikembangkan.

Tahap seleksi yang dilakukan adalah mengumpulkan berbagai komentar instagram yang memiliki permasalahan tentang penanganan banjir di Jakarta dengan link <https://www.instagram.com/aniesbaswedan/>. Seluruh komentar instagram tersebut kemudian diseleksi dan diubah menjadi dataset dengan format yang dapat diproses oleh model *machine learning*. Pelabelan polaritas sentimen dilakukan kepada seluruh observasi di *dataset* menurut makna teks komentar. Makna yang terdapat di *dataset* komentar yang diperoleh merupakan pendapat dan pandangan masyarakat terhadap penanganan banjir yang dilakukan oleh sekda DKI. Polaritas sentimen positif yang diberikan mengarah kepada lelucon atau ungkapan semangat kepada sekda DKI, sedangkan polaritas sentimen negatif yang diberikan mengarah kepada sindiran, ungkapan tidak setuju, dan ungkapan negatif. Pemakai bahasa terkadang menggunakan berbagai ungkapan untuk mengekspresikan kemarahan, kekesalan, kekecewaan, ketidaksenangan atau bahkan kebencian terhadap

suatu hal atau kejadian yang menyimpannya. Ungkapan tersebut sering disebut atau dikategorikan makian [11]. Tahap *preprocessing* yang dilakukan adalah membersihkan komentar instagram yang ada di dataset dan memberikan polaritas sentimen. Pembersihan dilakukan dengan cara melakukan *case folding*, *tokenizing*, dan *filtering*. Sedangkan pemberian polaritas sentimen dibedakan menjadi 1 yang berarti komentar positif dan 0 yang berarti komentar negatif. Gambar 2 menunjukkan dataset komentar instagram sebelum dan sesudah dilakukan tahap *preprocessing*. Berikut ini adalah tahap *preprocessing* yang dilakukan di dalam penelitian.

1. Case Folding

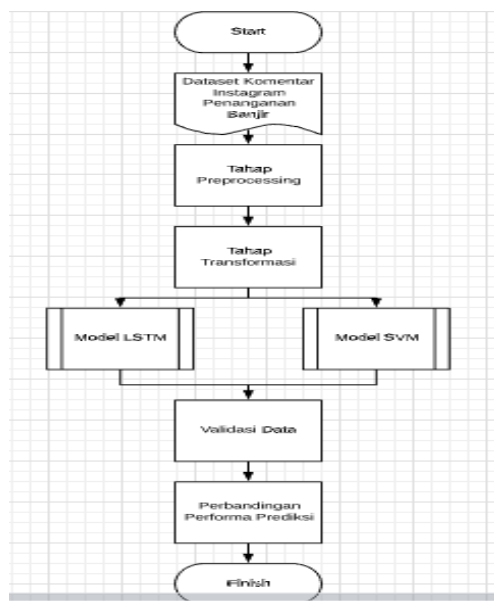
Case Folding adalah proses yang mengubah seluruh huruf besar menjadi huruf kecil.

2. Tokenizing

Tokenizing adalah proses memilah kata-kata yang ada di dokumen menjadi token.

3. Filtering

Filtering adalah proses penyaringan kata-kata yang ada di dokumen. Analisis yang dilakukan menggunakan algoritma *stopword* sehingga kata-kata yang tidak penting akan dihilangkan. Algoritma *stopword* juga menghilangkan tanda baca, angka, emoji, dan url.



Gambar 1 Flowchart Program

text	x1
1 Ayo doakan pak Anis dan jajarannya, yg berkerja denga...	ayo doakan pak anis dan jajarannya yg berkerja dengan ...
2 Nico devandri jaya	
3 Harusnya kembali kemasing2 induvidu. Bukan gubernur...	alhamdulillah semoga jakarta bebas banjir dan bpk cepe...
4 Pak Anies tolong keluarkan PerGub kalau buang sampa...	alhamdulillah smoga kami yg di kemayoran tdk kena ba...
5 Alhamdulillah pak anies, 🙏🙏❤️	sampe kapn psbb pak
6 Semangat pak!!! Semoga lekas pulihh❤️ Negara ini butu...	kami warga jakarta yg merasakan kerja bapak selalu men...
7 semoga lekas sembuh dan beraktivitas kembali pak gub...	ok boss banyak sholat malam bossancamanbanyak dari ...
8 Sehat2 Bpk2....	semangat dan sehat selalu pak anis salam dari medan
9 Terpaksa bener2 Terpaksa beberapa Pohon di Tebang.....	apakahbenar akan di bangun waduk di cipinang melayu...
10 @wayanayung @putubayus	psbb jngan diperpanjang lgi pak bngung dirumahkan tr...
11 Sehat terus ya pak barokalloh semoga Alloh selalu melin...	banyak cerita ttg mal fatih hampir putus asa menakluka...
12 🙏🙏🙏	wahh langgagganan sya pulau mas rawamgun kl hujan ...
13 !شكرا الحمد لله!	iya gitu kerjain rakyat tahu kok mana hasil kerja mana hs...

Gambar 2 Dataset Komentar Instagram Sebelum dan Sesudah Tahap *Preprocessing*

Tahap transformasi yang dilakukan adalah mengubah bentuk setiap observasi yang terdapat di dataset menjadi format yang dapat diproses oleh model algoritma *machine learning*. Komentar instagram sebagai observasi penelitian diubah menjadi bentuk vektor dan kelas *feature* polaritas sentimen diubah menjadi kelas target yang digunakan dalam klasifikasi biner sebagai penentuan persentase. Hal ini dilakukan dengan menggunakan *word embedding*, yaitu metode *TF-IDF* untuk *SVM* dan metode *Tokenizer* dari tensorflow keras untuk *RNN* tipe *LSTM*. *TF-IDF* adalah metode untuk mengetahui kata penting yang muncul dalam dokumen. Nilai *TF-IDF* dapat diketahui dengan menghitung *term frequency* dan *inverse-document frequency* dari kata yang akan dicari. Semakin mendekati nilai 1 maka nilai *TF-IDF* dari kata tersebut menyatakan bahwa kata tersebut penting dan merupakan *feature* yang baik untuk analisis. Sedangkan *tokenizer* adalah metode yang memproses teks dengan mengubahnya ke dalam bentuk vektor. Input yang diterima adalah *corpus* seluruh teks dan outputnya adalah vektor *feature* yang mempresentasikan kata-kata di dalam *corpus*. Input yang diterima dipecah menjadi *token* dan dilakukan *pad sequence* agar seluruh input memiliki ukuran yang sama. Input teks tersebut dikonversi menjadi *sequence* yang akan diproses menggunakan *one hot encoding*. *One hot encoding* merepresentasikan angka menjadi nilai vektor dimana semua elemen bernilai 0 kecuali hanya satu elemen yang bernilai 1 yaitu posisi elemen yang sama dengan kata pada kamus. Gambar 3 menunjukkan data latih dan data uji komentar instagram yang telah ditransformasi menjadi representasi vektor.

```

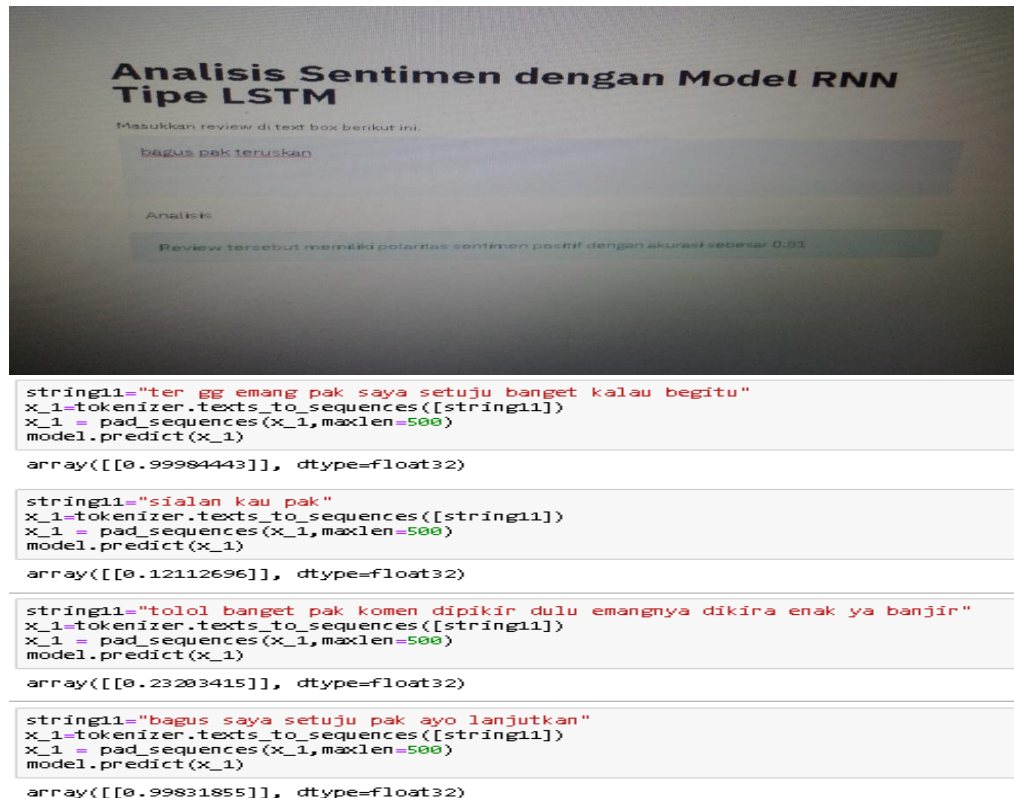
x_train
array([[ 0,  0,  0, ..., 617, 278, 18],
       [ 0,  0,  0, ...,  0,  0, 188],
       [ 0,  0,  0, ..., 4016, 728, 1608],
       ...,
       [ 0,  0,  0, ..., 18,  2, 1053],
       [ 0,  0,  0, ..., 2970, 11, 2971],
       [ 0,  0,  0, ...,  0, 16, 239]])

x_test
array([[ 0,  0,  0, ...,  0,  0, 2105],
       [ 0,  0,  0, ..., 28, 344, 169],
       [ 0,  0,  0, ...,  0, 4829, 87],
       ...,
       [ 0,  0,  0, ...,  2, 1623, 566],
       [ 0,  0,  0, ..., 1468, 323,  3],
       [ 0,  0,  0, ...,  5, 199,  4]])

```

Gambar 3 Representasi Vektor Dataset Komentar Instagram

Tahap *data mining* yang dilakukan adalah melakukan pelatihan kepada dataset menggunakan algoritma *machine learning*. Proses pelatihan yang dilakukan adalah mengubah nilai *weight* yang dimiliki setiap observasi apakah mereka termasuk ke dalam kelas target polaritas sentimen positif atau negatif. Tahap interpretasi dan evaluasi yang dilakukan adalah mengevaluasi performa model algoritma *machine learning* yang digunakan berdasarkan nilai *metric* yang dihasilkan. Nilai *metric* tersebut dapat berupa akurasi, presisi, *recall*, dan skor F1. Kemudian interpretasi dilakukan dengan cara melakukan prediksi terhadap data baru yang tidak berasal dari dataset yang digunakan. Gambar 4 menunjukkan implementasi prediksi terhadap data baru yang dilakukan di web framework streamlit. Semakin mendekati nilai 1 maka hasil prediksi menyatakan bahwa komentar tersebut bersentimen positif dan semakin mendekati 0 maka hasil prediksi menyatakan bahwa komentar tersebut bersentimen negatif.



Gambar 4 Implementasi Prediksi Analisis Sentimen

3. Hasil dan Pembahasan

Dalam penelitian yang dilakukan, dataset komentar instagram tentang penanganan banjir yang digunakan memiliki 1127 komentar negatif dan 1409 komentar positif. Dari seluruh komentar yang ada, 70% keseluruhan data digunakan sebagai data latih yaitu sebanyak 1775 dan 30% sisanya sebanyak 761 sebagai data testing. Data validasi komentar instagram yang digunakan kemudian dibandingkan oleh kuisioner yang diisi oleh ahli pakar bahasa Indonesia bernama Cicih Ratnasih, S.Pd.SD. Beliau memberikan polaritas sentimen kepada seluruh data validasi yang digunakan pada saat proses evaluasi. Kemudian *confusion matrix* yang dimiliki oleh model algoritma dengan hasil validasi ahli pakar bahasa dibandingkan. Hasil perbandingan ini menghasilkan bahwa model algoritma memiliki performa prediksi yang lebih baik dari hasil pengujian dengan ahli bahasa. Gambar 5 menunjukkan perbedaan *confusion matrix* model algoritma yang digunakan (kiri) dan hasil pengujian dengan ahli bahasa (kanan).

<pre> y_pred = model.predict(X_test) accuracy = accuracy_score(ytest, y_pred.round()) print('Accuracy: %f' % accuracy) # precision tp / (tp + fp) precision = precision_score(ytest, y_pred.round()) print('Precision: %f' % precision) # recall: tp / (tp + fn) recall = recall_score(ytest, y_pred.round()) print('Recall: %f' % recall) # f1: 2 tp / (2 tp + fp + fn) f1 = f1_score(ytest, y_pred.round()) print('F1 score: %f' % f1) </pre>					
<pre> Accuracy: 0.788436 Precision: 0.793478 Recall: 0.846868 F1 score: 0.819304 </pre>					
<pre> Confusion Matrix : [[65 238] [10 144]] Accuracy Score : 0.4573304157549234 Report : </pre>					
		precision	recall	f1-score	support
0	0.87	0.21	0.34	303	
1	0.38	0.94	0.54	154	
micro avg	0.46	0.46	0.46	457	
macro avg	0.62	0.57	0.44	457	
weighted avg	0.70	0.46	0.41	457	

Gambar 5 Confusion Matrix Model Algoritma yang Dihasilkan

Dalam struktur model algoritma RNN tipe LSTM diamati perubahan konfigurasi jumlah neuron pada setiap layer yang berbeda ataupun besar nilai *hyperparameter* agar diketahui dampaknya kepada performa prediksi. Layer-layer yang diamati adalah layer LSTM, layer *Dense*, dan layer *Dropout*. Untuk proses kompilasi yang dilakukan model pada layer output digunakan fungsi aktivasi *sigmoid*, fungsi loss *Binary Cross-Entropy*, optimasi tipe *adam*, dan *metrics accuracy*. *Binary Cross-Entropy Loss* adalah teknik yang mengukur seberapa jauh performa prediksi yang dilakukan model dalam nilai 0 dan 1. Optimasi tipe *adam* adalah algoritma optimasi dimana *learning rate* yang dilakukan model *deep learning* berubah secara adaptif sehingga setiap parameter memiliki learning rate individual. *Metrics accuracy* digunakan sebagai fungsi yang bertanggung jawab sebagai performa model, dimana pada penelitian yang dilakukan adalah akurasi prediksi model. Gambar 15 menunjukkan struktur model RNN tipe LSTM yang digunakan saat melakukan penelitian. Gambar tersebut menunjukkan salah satu observasi penelitian yang dilakukan menggunakan 25 neuron pada 1 layer LSTM, nilai layer *Dropout* sebesar 0,5, dan 1 neuron pada layer *Dense* sebagai layer output. Gambar 6 menunjukkan perbandingan performa model algoritma LSTM (kiri) dengan model algoritma SVM (kanan).



```

y_pred = model.predict(X_test)
accuracy = accuracy_score(ytest, y_pred.round())
print('Accuracy: %f' % accuracy)
# precision tp / (tp + fp)
precision = precision_score(ytest, y_pred.round())
print('Precision: %f' % precision)
# recall: tp / (tp + fn)
recall = recall_score(ytest, y_pred.round())
print('Recall: %f' % recall)
# f1: 2 tp / (2 tp + fp + fn)
f1 = f1_score(ytest, y_pred.round())
print('F1 score: %f' % f1)

```

```

import numpy as np
#lin_svm=svm.LinearSVC()
lin_svm= svm.SVC(kernel='linear',C=1)
lin_svm.fit(train_vectors, data_train['sentiment'])
#lin_svm.fit(test_vectors, data_test['sentiment'])
y_train_hat=lin_svm.predict(train_vectors)
y_test_hat=lin_svm.predict(test_vectors)

print( "train accuracy= " ,np.mean(y_train_hat == data_train['sentiment'])*100)
print( "test accuracy= " ,np.mean(y_test_hat == data_test['sentiment'])*100)

```

Accuracy: 0.788436
 Precision: 0.793478
 Recall: 0.846868
 F1 score: 0.819304

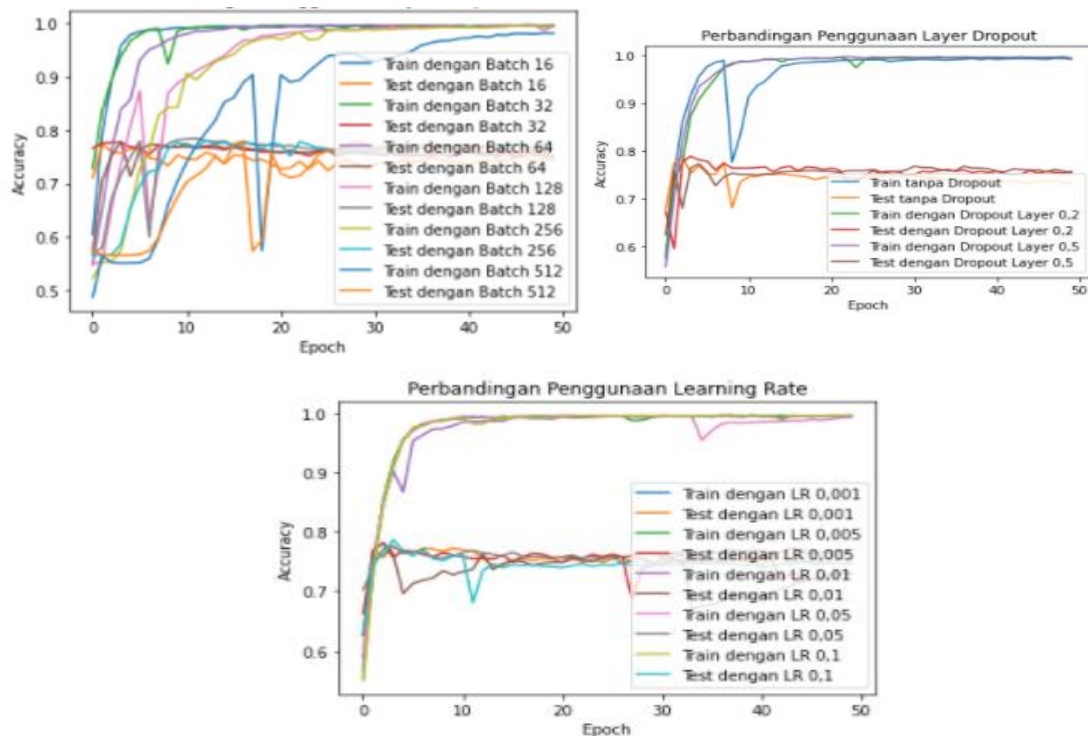
train accuracy= 95.43661971830986
 test accuracy= 73.98160315374507

Gambar 6 Perbandingan performa model algoritma LSTM dan VSM

Hasil penelitian membuktikan bahwa konfigurasi *hyperparameter* yang dilakukan kepada model algoritma yang digunakan mampu memperbaiki performa prediksi yang dihasilkan. Dari berbagai variasi jumlah layer sebanyak 1 dan 2 dengan 25, 50, 100, dan 200 neuron, diketahui bahwa 1 layer dan 25 neuron menghasilkan akurasi validasi yang paling baik. Performa yang dihasilkan memiliki titik optimum di 75% dan mampu mencapai titik global *maximum* di 78,84%. Hal ini ditunjukkan pada tabel 1. Konfigurasi *hyperparameter* lain yang dilakukan adalah variasi batch, penggunaan layer *dropout*, dan juga variasi besar *learning rate* yang digunakan. Performa paling baik yang dihasilkan adalah ukuran *batch* sebanyak 32, *dropout* layer sebesar 0,5, dan *learning rate* sebesar 0,00001. Hal ini dibuktikan oleh gradien yang dihasilkan selama proses *training* dan *testing*. Gambar 7 menunjukkan perbandingan performa konfigurasi *hyperparameter* model algoritma.

Tabel 1 Tabel Akurasi Validasi di Titik Global Maximum

Jumlah Layer	Jumlah Neuron	Besar Learning Rate		
		0,000001	0,001	1
2	200	57,82%	66,10%	56,64%
2	100	57,16%	75,56%	56,64%
2	50	57,69%	75,82%	56,64%
2	25	57,29%	77,40%	56,64%
1	200	60,05%	75,03%	60,32%
1	100	58,74%	75,30%	60,71%
1	50	61,37%	77,00%	55,72%
1	25	61,37%	78,84%	58,87%



Gambar 7 Perbandingan Performa Konfigurasi *Hyperparameter* Model Algoritma

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa model algoritma RNN tipe LSTM memiliki performa yang lebih baik daripada model algoritma SVM. Model LSTM memiliki akurasi sebesar 78,84%, presisi sebesar 79,34%, *recall* sebesar 84,68%, dan skor F1 sebesar 81,93%. Sedangkan model SVM hanya memiliki akurasi sebesar 73,98%, presisi sebesar 69,5%, *recall* sebesar 57,5%, dan skor F1 sebesar 53%. Performa yang dimiliki model LSTM juga lebih baik pada fase training. Model LSTM mampu mencapai 99%, sedangkan model SVM hanya dapat mencapai 93%.

Penelitian yang dilakukan masih memiliki banyak kekurangan. Model yang dilatih terjadi *overfit* sehingga performa prediksi terhadap data baru yang tidak berasal dari data training akan menghasilkan prediksi yang kurang akurat. Hal ini dibuktikan dengan nilai akurasi yang meningkat dan nilai loss yang menurun pada fase training, namun nilai akurasi stagnan dan nilai *loss* semakin menaik pada fase validasi.

Pada penelitian ini terdapat beberapa hal yang dapat dikembangkan untuk penelitian selanjutnya agar menghasilkan performa prediksi yang lebih baik. Salah satunya adalah dengan melakukan interpolasi data sehingga dataset komentar yang digunakan menghasilkan gradien yang stabil. Hal lainnya adalah melakukan proses *stemming* dan *speech tagging* dalam tahap preprocessing. Dari segi implementasi, penelitian ini merupakan klasifikasi biner, sehingga penelitian klasifikasi *multi-class* dengan tiga jenis polaritas sentimen atau lebih dapat dilakukan sebagai perbandingan penelitian.

Referensi

- [1] Ahmad, Munir, Shabib Aftab, dan Iftikhar Ali. 2017. Sentiment Analysis of Tweets using SVM. *International Journal of Computer Applications*. 177(5): 25-29.
- [2] Colace, Francesco, Massimo DS, dan Luca Greco. 2014. SAFE: A Sentiment Analysis Framework for E-Learning. *IJET*. 9(6): 37-41.

- [3] Gunawan, Billy, Helen SP, dan Enda EP. 2018. Sistem Analisis Sentimen pada Ulasan Produk Menggunakan Metode Naïve Bayes. *Jurnal Edukasi dan Penelitian Informatika*. 4(2): 113-118.
- [4] Ivanedra, Kasyfi dan Metty Mustikasari. 2019. Implementasi Metode Recurrent Neural Network pada Text Summarization dengan Teknik Abstraktif. *Jurnal Teknologi Informasi dan Ilmu Komputer*. 6(4): 377-382.
- [5] Jianqiang, Zhao, Gui Xiaolin, dan Zhang Xuejun. 2018. Deep Convolution Neural Networks for Twitter Sentiment Analysis. *IEEE*. 6: 23253-23260.
- [6] Lutfi, AA, Adhistya EP, dan Silmi Fauziati. 2018. Sentiment Analysis in the Sales Review of Indonesian Marketplace by Utilizing Support Vector Machine. *JISEBI*. 4(1): 57-64.
- [7] Nurrohmat, MA dan Azhari SN. 2019. Sentiment Analysis of Novel Review using Long Short Term Memory Method. *IJCCS*. 13(3): 209-218.
- [8] Octaviani, PA, Yuciana Wilandari, dan Dwi Ispriyanti. 2014. Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. *Gaussian*. 3(4): 811-820.
- [9] Rashid, TA, Polla Fattah, dan Delan KA. 2018. Using Accuracy Measure for Improving the Training of LSTM with Metaheuristic Algorithms. *Procedia Computer Science*. 140: 324-333.
- [10] Thomas, Merin dan Latha CA. 2014. Sentiment Analysis using Recurrent Neural Network. *IJET*. 7(2,27): 88-92.
- [11] Zulfa, Ira dan Edi Winarko. 2017. Sentimen Analisis Tweet Berbahasa Indonesia dengan Deep Belief Network. *Indonesian Journal of Computing and Cybernetics Systems*. 11(2): 187-198.
- [12] Aldi, MWP, Jondri, dan Annisa Aditsania. *Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin*. e-Proceeding of Engineering. Bandung: 2018; 5(2): 3548-3555.
- [13] Patel, Alpna dan Arvind KT. *Sentiment Analysis by using Recurrent Neural Network*. Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE). Sultanpur: 2019.