

Klasifikasi Ekspresi Teks Berbahasa Jawa Menggunakan Algoritma Long Short Term Memory

Oddy Virgantara Putra¹, Aziz Musthafa², Kukuh Prasetyo Wibowo^{3*}

^{1),2),3)}Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Darussalam Gontor
Jl. Raya Siman km 6, Ponorogo, Indonesia 63471

*email: kukuhprasetyo@mhs.unida.gontor.ac.id

(Naskah masuk: tgl. 26 Februari 2021; diterima untuk diterbitkan: 19 Maret 2021)

ABSTRAK – Suku Jawa adalah suku terbesar yaitu 41% atau sekitar 95.217.022 jiwa. Pulau Jawa juga pengakses internet terbesar khususnya untuk media sosial, yaitu 87,13%. Dari data tersebut banyak informasi yang bisa didapatkan dari keaktifan suku Jawa dalam menggunakan media social, khususnya adalah data tekstual. Tetapi tidak mudah mendapatkan ekspresi atau emosi suku Jawa dari media sosial karena jumlah datanya sangat banyak. Penelitian ini bertujuan untuk mengklasifikasikan ekspresi teks bahasa Jawa khususnya dalam media sosial yang menggunakan bahasa Jawa dan menguji keakuratan algoritma long short term memory (LSTM) dalam mengklasifikasikan ekspresi teks bahasa Jawa berdasarkan jenis ekspresinya. Penelitian ini juga membandingkan dua algoritma lainnya, yaitu support vector machine (SVM) dan random forest (RF). Metode yang digunakan adalah text mining yang sudah banyak digunakan untuk kebutuhan machine learning dalam pengolahan data tekstual. Bahasa pemrograman yang dipakai adalah Python dan code editor yang dipakai, yaitu google colabs serta google drive sebagai perangkat untuk menyimpan dataset. Hasil proses penggunaan LSTM dan perbandingannya dengan SVM dan RF untuk memilah ekspresi menjadi empat ekspresi, yaitu marah, senang, sedih, dan takut. LSTM mendapat akurasi 92%, akurasi tersebut menunjukkan LSTM efektif dalam pengklasifikasian ekspresi teks berbahasa Jawa karena mendapat akurasi paling tinggi.

Kata Kunci – Text Mining, Bahasa Jawa, Ekspresi, Long Short Term Memory.

Classification of Javanese Text Expressions Using Long Short Term Memory Algorithm

ABSTRACT – The Javanese are the largest tribe of 41% or about 95,217,022 people. Java island is also the largest internet access, especially for social media, which is 87.13%. From the data a lot of information can be obtained from the activeness of Javanese people in using social media, especially textual data. But, it is not easy to get Javanese expressions or emotions from social media because the amount of data is very large. This study aims to classify the expression of Javanese text, especially in social media that uses Javanese language and test the accuracy of long short term memory (LSTM) algorithm in classifying Javanese text expression based on the type of expression. This study also compares two other algorithms namely support vector machine (SVM) and random forest (RF). The method used is text mining that has been widely used for machine learning needs in textual data processing. The programming language used is Python and the code editor used is google colabs and google drive as tools to store datasets. From the results of the process of using LSTM and comparing with SVM and RF to sort the expression into four expressions namely angry, happy, sad, and afraid. LSTM has 92% accuracy, which indicates that LSTM is effective in classifying Javanese text expressions because it has the highest accuracy.

Keywords - Text Mining, Javanese, Expression, Long Short Term Memory.

1. PENDAHULUAN

Suku Jawa adalah suku terbesar yang ada di

Indonesia. Dengan ini suku Jawa sangat berpotensi sebagai pengguna media sosial terbesar yang ada di Indonesia[1]. Pulau Jawa

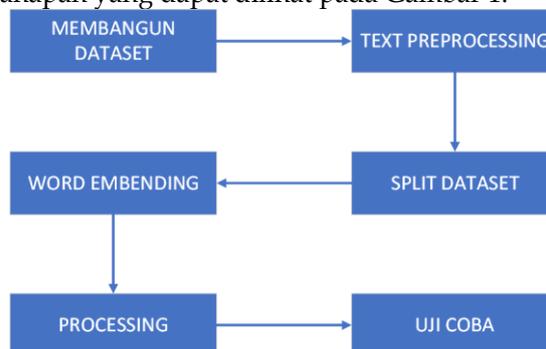
juga mempunyai kontribusi besar atas penetrasi internet di Indonesia yang mencapai angka penetrasi 55% dari total keseluruhan yang dikutip dari website Kompas.com. Kominfo juga melakukan survei yang menunjukkan 87,13% pengguna internet di Indonesia digunakan untuk mengakses situs media sosial. Dari presentasi data tersebut maka banyak sekali informasi yang dapat dimanfaatkan seperti melihat opini publik tentang suatu produk atau dipakai oleh pengembang media sosial untuk melihat emosi atau ekspresi pengguna[2]. Namun, dalam mencari ekspresi dari suatu data sangat tidak mudah karena jumlahnya sangat banyak. Selain itu juga, suku Jawa sangat susah ditebak ekspresinya karena mempunyai sifat sopan dan sedikit pemalu[3]. Maka dari itu dalam penelitian ini peneliti mengusulkan untuk menggunakan *teks mining* sebagai cara untuk mengklasifikasikan teks media sosial berbahasa Jawa.

Penelitian terkait dengan ini adalah penelitian yang dilakukan oleh Oddy Virgantara Putra, Shofin Nahfwa Utama, dan Fathin Muhammad Wasmanson yang meneliti tentang emosi teks pada media sosial dengan menggunakan bahasa Sunda. Algoritme yang dipakai yaitu *support vector machine* (SVM) dan mengklasifikasikan emosi menjadi empat ekspresi, yaitu marah, senang, sedih, dan takut. Hasil akurasi yang diperoleh adalah 95,6% dengan jumlah dataset sebesar 2000 data yang seluruhnya diambil dari media sosial.

Perbedaan dengan penelitian terdahulu adalah objeknya. Belum ada penelitian yang memakai bahasa Jawa sebagai objeknya. Dalam penelitian ini bertujuan untuk membuat model pengklasifikasian data tekstual berbahasa Jawa dengan memakai tiga algoritme, yaitu *Long Short Term Memory* (LSTM), *Support Vector Machine* (SVM) dan *Random Forrest* (RF). Ketiga algoritme ini adalah algoritme yang sering dipakai untuk pengklasifikasian khususnya adalah klasifikasi teks[4]. Ketiga algoritme ini juga akan sekaligus dibandingkan dengan menggunakan dataset yang sama dalam pembuatan model teks berbahasa Jawa untuk mengetahui akurasi yang paling tinggi. Proses pengklasifikasian secara umum, yaitu *preparing data*, *text preprocessing*, *processing*, dan *confusion matrix*. Penelitian mengklasifikasi dalam empat ekspresi, yaitu marah, senang, sedih, dan takut.

2. METODE DAN BAHAN

Tahapan penelitian ini mempunyai beberapa tahapan yang dapat dilihat pada Gambar 1.



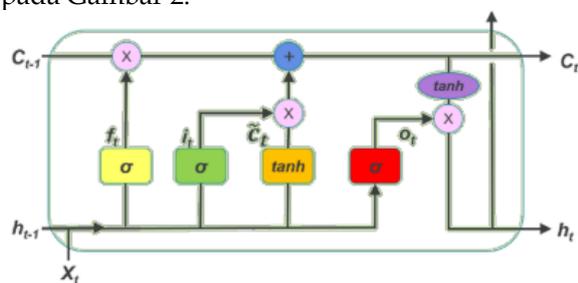
Gambar 1. Tahapan Penelitian

Pada Gambar 1 diperlihatkan tahapan penelitian yang pertama adalah membangun dataset yang diambil dari media sosial *facebook* dan *twitter* yang kemudian dilabeli dan divalidasi. Kemudian *text preprocessing* dalam tahap ini dataset akan dibersihkan sehingga data siap untuk diproses. Setelah data bersih maka data akan dibagi atau *split dataset* di mana pembagian akan dibagi dengan perbandingan 80:20 sebagai data *training* dan data *test*. Masuk ke dalam ekstraksi data yaitu *word embedding* data akan diekstraksi menjadi bentuk *array* yang kemudian akan diproses dengan membandingkan tiga algoritme, yaitu LSTM, SVM, dan RF. Setelah itu akan diuji menggunakan *confusion matrix* untuk melihat mana algoritme yang akurasi yang paling tinggi.

Metode yang dipakai adalah *text mining* dengan memakai tiga algoritme yang akan dibandingkan untuk mengetahui algoritme yang akurasi yang paling tinggi sehingga efektif untuk dipakai sebagai model pengklasifikasian teks berbahasa Jawa.

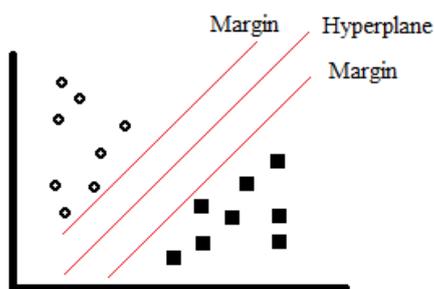
Algoritme yang pertama, LSTM, dapat mengingat informasi jangka panjang. Pada *Recurrent Neural Network* (RNN), perulangan jaringan hanya menggunakan satu layer sederhana, yaitu layer tanh[5]. LSTM memiliki sel memori yang terdiri dari empat komponen utama: input gate, koneksi berulang, forget gate dan output gate. Input gate dapat memblokir atau memungkinkan sinyal masuk untuk mengubah kondisi sel[6]. Demikian pula, output gate baik mencegah atau memungkinkan keadaan sel dari memiliki efek pada unit lain[7]. *Forget gate* memungkinkan sel untuk mengingat atau melupakan keadaan sebelumnya dengan

mengontrol koneksi berulang-ulang sel[8]. Contoh struktur jaringan LSTM diperlihatkan pada Gambar 2.



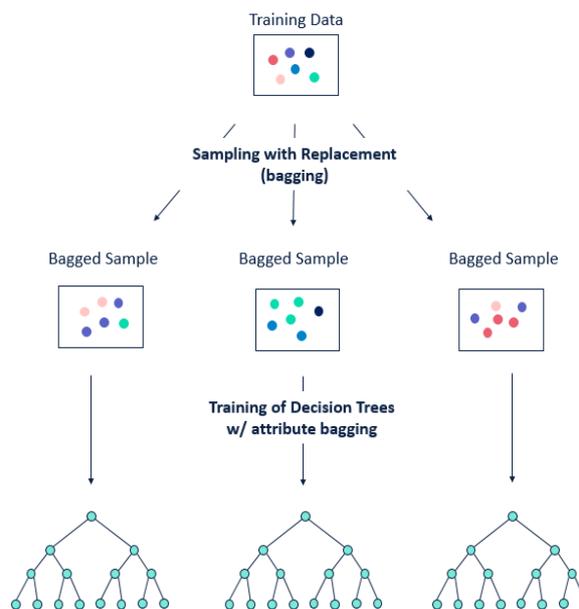
Gambar 2. Struktur Jaringan LSTM

Algoritme yang kedua, SVM (SVM), adalah algoritme *machine learning* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada input *space*[9]. Metode dari SVM biasanya digunakan untuk melakukan klasifikasi dengan masalah yang linier, namun dalam perkembangannya metode SVM dapat digunakan untuk permasalahan yang non-linear dengan cara mencari *hyperplane* yang digunakan untuk jarak yang maksimal antar kelas data[10]. Untuk menyelesaikan problem non-linear, SVM dimodifikasi dengan memasukkan fungsi Kernel[11]. Ada beberapa jenis kernel yang ada dapat digunakan dalam SVM, yaitu Polynomial, Linear, Gaussian RBF dan Sigmoid[12]. Contoh pemetaan SVM dapat dilihat pada Gambar 3.



Gambar 3. Pemetaan SVM

Algoritme yang ketiga, *Random Forest* (RF), adalah pengembangan dari metode CART (*Classification and Regression Tree*), yaitu dengan menerapkan metode *bootstrap aggregating* (*bagging*) dan *random feature selection*[13]. Dalam RF, banyak pohon ditumbuhkan sehingga terbentuk hutan (*forest*), kemudian analisis dilakukan pada kumpulan pohon tersebut[14]. Contoh penerapannya dapat dilihat pada Gambar 4.



Gambar 4. Training menggunakan RF

Bahan yang digunakan dalam penelitian yaitu *Microsoft Excel* yang dipakai untuk menampung dataset yang berbentuk tekstual, media sosial Twitter dan Facebook digunakan untuk mengumpulkan data tekstual berbahasa Jawa, *google colabs* untuk tempat program dijalankan (aplikasi ini tidak memerlukan instalasi *library* atau aplikasi sehingga sangat mudah digunakan dimana saja dan kapan saja tetapi kekurangannya adalah harus terkoneksi internet), dan yang terakhir adalah *google drive* walaupun dataset yang digunakan hanya satu file tetapi *google drive* ini berguna sebagai penampung data sehingga dataset nanti akan dipanggil melalui *google colabs*[15].

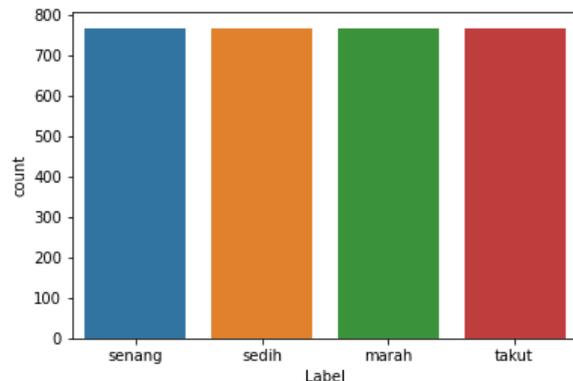
3. HASIL DAN PEMBAHASAN

3.1. Hasil

a. Membangun Dataset

Tahap ini mengumpulkan dataset menggunakan dua media sosial, yaitu Facebook dan Twitter. kemudian data tersebut akan dilabeli menjadi empat ekspresi, yaitu marah, senang, sedih, dan takut. Setelah itu, data akan divalidasi apakah data yang sudah terlabeli benar atau salah. Kemudian data akan dibenarkan oleh ahli bahasa Jawa (ahli bahasa Jawa di sini adalah guru bahasa Jawa yaitu Bapak Muhammad Khoirudi, S.Pd yang mengajar di SMA Negeri 02 Ponorogo). Dataset yang terkumpul dalam penelitian ini

berjumlah 3072 data dan sudah tervalidasi. Selain itu, data akan disamaratakan sehingga nantinya data akan berimbang. Hasil dataset dapat dilihat pada Gambar 5.



Gambar 5. Identitas Dataset

b. *Text Preprocessing*

• *Case Folding*

Pada tahap pertama *text preprocessing* adalah *case folding* dimana dataset akan dibagi menjadi dua, kolom pertama pada data adalah label dan kolom kedua adalah fitur. Fitur inilah yang kemudian akan diproses, dimana dalam *case folding* ini mengubah fitur menyamaratakan menjadi huruf kecil semua sehingga mudah diproses. Contoh *case folding* dapat dilihat pada Tabel 1.

Tabel 1. Contoh penerapan *case folding*

Sebelum Case Folding	Sesudah Case Folding
Wong lanang ki pie to karepe, Tuluooong :)	wong lanang ki pie to karepe, tuluooong :)
Tapi asli kenyataane iki cen BAJINGAN TENAN! Ndem gheden!	tapi asli kenyataane iki cen bajingan tenan! ndem gheden!
Kabeh status e kok sayang sayangan to.... 🙄🙄	kabeh status e kok sayang sayangan to.... 🙄🙄
Tanpo mbok sadari sikapmu kwi nglarani aku :)	tanpo mbok sadari sikapmu kwi nglarani aku :)
Loro cok rasane ki Kenyataane ncen BAJINGAN	loro cok rasane ki kenyataane ncen bajingan

• *Filtering*

Tahap *filtering* menghilangkan karakter yang tidak diperlukan seperti simbol, nomor, dan karakter yang tidak penting dalam fitur. Contoh penerapan tersebut dapat dilihat pada Tabel 2.

Tabel 2. Contoh penerapan *filtering*

Sebelum Filtering	Sesudah Filtering
wong lanang ki pie to karepe, tuluooong :)	wong lanang ki pie to karepe tuluooong
tapi asli kenyataane iki cen bajingan tenan! ndem gheden!	tapi asli kenyataane iki cen bajingan tenan ndem gheden
kabeh status e kok sayang sayangan to.... 🙄🙄	kabeh status e kok sayang sayangan to
tanpo mbok sadari sikapmu kwi nglarani aku :)	tanpo mbok sadari sikapmu kwi nglarani aku
loro cok rasane ki kenyataane ncen bajingan	loro cok rasane ki kenyataane ncen bajingan

• *Stopword Removal*

Stopword removal merupakan proses menghilangkan kata yang tidak diperlukan dari fitur seperti kata sambung, singkatan, dan kata yang tidak mempunyai makna sehingga tidak membantu dalam proses pengklasifikasian. Penghapusan kata tersebut juga bertujuan untuk mengurangi jumlah kata dalam fitur yang nantinya bisa berpengaruh dalam proses NLP (*Natural Language Processing*). Beberapa kata yang akan dihapus dalam penelitian ini dapat dilihat pada Tabel 3 dan contoh penerapannya pada Tabel 4.

Tabel 3. Beberapa kata yang dihapus

Daftar kata yang dihapus
'seng'sing'akeh'ameh'uwes'tok'kudu'bedo 'nggo'nyaur'sampean'ono'ngroso'lali'etok 'ben'nang'ati'luweh'bocah'minder'dancok 'jancok'asu'su'raimu'koyok'matamu'mripatmu 'celeng'pas'ngeyel'dadi'yowis'kroso'Ngene'iki 'podo'westo'wisto'cen'nganti'entek'ndueni'urep 'mergo'pas'iri'nek'kui'ndak'ndas'bos'sue'due 'duwe'ngopo'do'penak'dewe'ngapusi'urung 'durung'rung'apal'hapal'hafal'elek'ganteng'ayu 'cilik'gede'dowo'pendek'los'up'story'nanging 'semangat'anak'ngerti'postingane'wani'jantung 'tanpo'tapi'lelakon'mbah'dhuwe'arti'iki'tak'pek 'suargo'urusane'gugu'ninggal'teko'mangkat 'kenopo'mumor'jungker'walek'wajan'kosok 'jujur'angker'bersih'resik'karo'filosofine'neroko 'surago'sego'goreng'ajor'ngedoh'rempelo'seneng 'sedih'nlongso'kecoblos'jajan'kang'turu'tangi 'saben'mung'salah'bener'luput'nyatane'moto 'moco'ngurus'ndrodok'isine'gegerke'ijek'taseh 'bien'ndisek'goblok'sekirane'mbe'dino'berarti 'gendak'mantu'edan'setia'malah'online'rame'sepi 'pilih'milih'lungo'budal'pamit'cuek'gelem'seko'm 'bien'ndadak'ngguyu'nangis'kekel'tetep'tuku'bakar

Tabel 4. Contoh penerapan stopword removal

Sebelum Stopword Removal	Sesudah Stopword Removal
wong lanang ki pie to karepe tuluooong	wong lanang karepe
tapi asli kenyataane iki cen bajingan tenan ndem gheden	asli kenyataane bajingan tenan
kabeh status e kok sayang sayangan to	kabeh kok sayang sayang
tanpo mbok sadari sikapmu kwi nglarani aku	tanpo mbok sadari nglarani aku
loro cok rasane ki kenyataane ncen bajingan	loro cok rasane kenyataane bajingan

• *Tokenizing*

Tahap terakhir dari proses *text processing* adalah *tokenizing* yaitu melakukan pemisahan antara setiap kata. Pada proses ini kalimat akan dipisahkan perkata menggunakan simbol pemisah seperti petik dua ("), tanda strip (-), dan juga bisa menggunakan tanda *underscore* (_). Contoh *tokenizing* dapat dilihat pada Tabel 5.

Tabel 5. Contoh penerapan tokenizing

Sebelum Tokenizing	Sesudah Tokenizing
wong lanangkarepe	wong-lanang-karepe
asli kenyataane bajingan tenan	asli-kenyataane-bajingan-tenan
kabeh kok sayang sayang	kabeh-kok-sayang-sayang
tanpo mbok sadari nglarani aku	tanpo-mbok-sadari-nglarani-aku
loro cok rasane kenyataane bajingan	loro-cok-rasane-kenyataane-bajingan

c. *Processing*

Proses ketiga algoritme ini tidak terlalu beda jauh hanya beberapa yang berbeda antar satu dan lainnya.

• Proses *LSTM*

Pada proses *LSTM* menggunakan 150 *neuron* yang kemudian semua jaringan dihubungkan ke setiap *neuron* dengan jumlah berdasarkan banyaknya kelas pada penelitian ini. Tahap selanjutnya, adalah penggunaan aktivasi fungsi sigmoid. Setelah semua dibangun, model akan dikonfigurasi terlebih dahulu dengan menggunakan optimasi *Adam* untuk

mengetahui nilai *loss* dari model yang sudah dibuat. Selanjutnya, parameter untuk membantu proses *training* adalah ukuran *batch size* dan *epoch*. Parameter yang didapatkan sebesar 557,256.

• Proses *SVM*

Proses *SVM* dibantu dengan metode pengklasifikasian multikelas, yaitu metode *OvO (One versus One Classification)*, karena dasar algoritme *SVM* ini diperuntukan hanya untuk pengklasifikasian dua kelas saja. Pada proses ini juga dibantu dengan menggunakan kernel linier. Dalam proses *OvO*, proses *training* akan menjadi enam kali, karena objek pengklasifikasian dalam penelitian ini adalah empat kelas dan semua kelas akan dipertemukan satu per satu. Hasil dari *OvO* ini adalah hasil terbanyak dari prediksi enam kali test tersebut.

• Proses *RF*

Proses *RF* adalah melatih dataset sehingga data akan dibuat menjadi model untuk pengklasifikasian. Setelah model terbentuk, model akan diuji sehingga akan didapatkan hasil dari prediksinya. Proses pengklasifikasian juga didukung dengan penggunaan *estimators*.

d. Uji Coba

Dalam penelitian ini, metode *confusion matrix* digunakan untuk menganalisa kinerja dan akurasi dari model yang sudah dibuat. *Confusion matrix* akan membandingkan hasil dari prediksi terhadap hasil klasifikasi yang sebenarnya. Ketiga lagoritma ini mendapatkan hasil *confusion matrix* yang berbeda-beda seperti ditampilkan pada Tabel 6, Tabel 7, dan Tabel 8.

Tabel 6. *Confusion matrix LSTM*

Accuracy	92%			
	Precision	Recall	F1-score	support
marah	91%	95%	93%	128
sedih	93%	92%	93%	125
senang	97%	90%	94%	125
takut	89%	93%	91%	126

Tabel 7. *Confusion matrix SVM*

Accuracy	50%			
	Precision	Recall	F1-score	support
marah	0%	0%	0%	1
sedih	0%	0%	0%	1
senang	100%	100%	100%	1
takut	33%	100%	50%	1

Tabel 8. *Confusion matrix RF*

Accuracy	27%			
	Precision	Recall	F1-score	support
marah	30%	23%	36%	164
sedih	29%	52%	37%	183
senang	23%	9%	13%	129
takut	25%	19%	22%	139

3.2. Pembahasan

Dari *training* hasil *confusion matrix* yang dilakukan ketiga algoritme ini didapatkan satu algoritme yang mendapat nilai akurasi paling tinggi, yaitu *LSTM* sebesar 92% dan ada algoritme yang mendapat nilai akurasi yang rendah yaitu *RF* sebesar 27%.

Dalam hal ini *SVM* mendapat nilai akurasi sebesar 50% tetapi presisi, recall, f1-score, dan support pada ekspresi senang mendapat nilai sempurna yang yaitu 100% berbeda sedikit dengan *LSTM* yang hampir semua mendapat nilai bagus rata-rata di atas 90%.

RF yang sedikit berbeda mendapat nilai presisi, recall, f1-score, dan support di semua ekspresi dengan nilai paling rendah tetapi hanya berbeda sedikit dengan *Support Vector Machine*. Dari ketiga algoritme memberikan nilainya tersendiri dan ini yang menjadi keunikan dalam penelitian ini.

Dalam penelitian tidak memakai stemming karena pengaturan dalam Bahasa Jawa khususnya dalam bentuk kalimat, ketika kata yang berimbuhan dijadikan kata dasar akan sedikit mengubah arti kata tersebut.

4. KESIMPULAN

Algoritme *LSTM* dapat digunakan secara efektif karena mendapat akurasi yang paling tinggi dalam pengklasifikasian ekspresi teks berbahasa Jawa dari media sosial, dengan akurasi sebesar 92%. Dengan membandingkan 3 algoritme yaitu *LSTM*, *SVM*, dan *RF* didapatkan hasil menggunakan algoritme *LSTM* paling bagus

dengan akurasi 92% dibandingkan dengan algoritme *SVM* dan *RF* dengan hasil akurasi 50% dan 27%.

5. DAFTAR PUSTAKA

- [1] W. Wahyudiyono, "Implikasi Penggunaan Internet Terhadap Partisipasi Sosial di Jawa Timur," *J. Komunika: Jurnal Komunikasi, Media dan Informatika*, vol. 8, no. 2, pp. 63–68, 2019, doi: 10.31504/komunika.v8i2.2487.
- [2] M. Maulana, *Deteksi Emosi Manusia pada Tweet Bahasa Indonesia dengan Klasifikasi Naive Bayes*. Disertasi, Institut Teknologi Sepuluh Nopember, 2016.
- [3] A. P. Kurniawan and N. U. Hasanat, "Perbedaan ekspresi emosi pada beberapa tingkat generasi suku di Yogyakarta," *J. Psikol.*, vol. 34, no. 1, pp. 1–17, 2007.
- [4] D. A. Muthia, "Komparasi Algoritma Klasifikasi Text Mining untuk Analisis Sentimen Pada Review Restoran," *J. PILAR Nusa Mandiri*, vol. 14, no. 1, pp. 69–74, 2018.
- [5] L. Wiranda and M. Sadikin, "Penerapan Long Short Term Memory pada Data Time Series untuk Memprediksi Penjualan Produk PT. Metiska Farma," *J. Nas. Pendidik. Tek. Inform.*, vol. 8, no. 3, pp. 184–196, 2019.
- [6] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM Neural Network for Text Classification," 2015, [Online]. Available: <http://arxiv.org/abs/1511.08630>.
- [7] M. A. Nurrohmat and A. SN, "Sentiment Analysis of Novel Review Using Long Short-Term Memory Method," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 13, no. 3, p. 209, 2019, doi: 10.22146/ijccs.41236.
- [8] W. K. Sari, D. P. Rini, and R. F. Malik, "Text Classification Using Long Short-Term Memory With GloVe Features," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 5, no. 2, p. 85, 2020, doi: 10.26555/jiteki.v5i2.15021.
- [9] O. V. Putra, F. M. Wasmanson, T. Harmini, and S. N. Utama, "Sundanese Twitter Dataset for Emotion Classification," *CENIM 2020 - Proceeding Int. Conf. Comput. Eng. Network, Intell. Multimed.* 2020, no. November, pp. 391–395, 2020, doi: 10.1109/CENIM51130.2020.9297929.
- [10] I. M. D. Ardiada, M. Sudarma, and D. Giriantari, "Text Mining pada Sosial Media untuk Mendeteksi Emosi Pengguna Menggunakan Metode Support Vector

- Machine dan K-Nearest Neighbour," *Maj. Ilm. Teknol. Elektro*, vol. 18, no. 1, p. 55, 2019, doi: 10.24843/mite.2019.v18i01.p08.
- [11] M. F. Fibrianda and A. Bhawiyuga, "Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer dengan Metode Naïve Bayes dan Support Vector Machine (SVM)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, pp. 3112–3123, 2018.
- [12] J. Mase, M. T. Furqon, and B. Rahayudi, "Penerapan Algoritme Support Vector Machine (SVM) pada Pengklasifikasian Penyakit Kucing," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 10, pp. 3648–3654, 2018.
- [13] I. M. Budi Adnyana, "Prediksi Lama Studi Mahasiswa dengan Metode Random Forest (Studi Kasus : Stikom Bali)," *CSRID (Computer Sci. Res. Its Dev. Journal)*, vol. 8, no. 3, pp. 201–208, 2016, doi: 10.22303/csrid.8.3.2016.201-208.
- [14] S. Almouzini, M. Khemakhem, and A. Alageel, "Detecting Arabic Depressed Users from Twitter Data," *Procedia Comput. Sci.*, vol. 163, pp. 257–265, 2019, doi: 10.1016/j.procs.2019.12.107.
- [15] Y. D. Rosita and Y. N. Sukmaningtyas, "LSTM Network and OCR Performance for Classification of Decimal Dewey Classification Code," *Rec. Libr. J.*, vol. 6, no. 1, p. 45, 2020, doi: 10.20473/rlj.v6-i1.2020.45-56.