

Analisis Sentiment Calon Presiden Indonesia 2019 dengan Metode Support Vector Machines dan Long-Short Term Memory Pada Media Sosial Twitter

Andika Pratama

School of Electrical Engineering

Telkom University

Bandung, Indonesia

andikatama@student.telkomuniversity.ac.id

Panji Christoper Silalahi

School of Electrical Engineering

Telkom University

Bandung, Indonesia

Panjichrst@student.telkomuniversity.ac.id

Abstrak - Sentiment analysis banyak digunakan oleh pemangku kepentingan untuk menilai sentimen terhadap suatu objek. Penelitian ini fokus pada analisis sentimen terhadap tokoh politik, khususnya calon presiden 2024, yang ramai dibicarakan oleh netizen di Twitter. Permasalahan yang diangkat dalam penelitian ini adalah kinerja algoritma dalam melakukan klasifikasi sentimen, di mana beberapa algoritma sering kali memiliki tingkat akurasi yang rendah. Tujuan penelitian ini adalah untuk meningkatkan ukuran kinerja dari penelitian sebelumnya yang menggunakan algoritma Naïve Bayes yang diketahui memiliki akurasi yang relatif rendah. Dalam penelitian ini, algoritma Support Vector Machine (SVM) dan Long-Short Term Memory (LSTM) digunakan untuk meningkatkan ukuran kinerja tersebut.

Keyword: Presiden, Politik, Algoritma

I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang pesat telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam ranah politik. Media sosial, khususnya Twitter, telah menjadi platform utama bagi masyarakat untuk mengekspresikan opini, pandangan, dan sentimen mereka terhadap berbagai isu, termasuk politik. Dalam konteks pemilihan presiden, opini dan sentimen publik yang diungkapkan melalui Twitter dapat memberikan wawasan yang berharga bagi pemangku kepentingan, seperti tim kampanye, partai politik, dan calon presiden itu sendiri.

Analisis sentimen adalah teknik yang digunakan untuk menentukan sikap atau perasaan seseorang terhadap suatu subjek berdasarkan teks yang dihasilkan. Dalam penelitian ini, analisis

sentimen digunakan untuk mengidentifikasi dan mengklasifikasikan sentimen masyarakat terhadap calon presiden Indonesia pada pemilihan tahun 2019. Mengingat besarnya volume data yang dihasilkan di Twitter, teknik analisis otomatis menjadi sangat penting.

Algoritma Naïve Bayes telah banyak digunakan dalam analisis sentimen sebelumnya, namun memiliki keterbatasan dalam hal akurasi. Untuk mengatasi hal ini, penelitian ini akan menggunakan dua algoritma yang lebih canggih, yaitu Support Vector Machines (SVM) dan Long-Short Term Memory (LSTM). SVM dikenal efektif dalam klasifikasi teks, sementara LSTM, yang merupakan jenis jaringan saraf tiruan (neural network), mampu menangani data sekuensial dengan lebih baik.

II. PEKERJAAN TERKAIT

Pada penelitian sebelumnya yang dilakukan Muhammad Raihan Fais Sya'bani mengenai Analisis Sentimen Terhadap Bakal Calon Presiden 2024 dengan Algoritma Naïve Bayes telah dilakukan analisis sentimen menggunakan metode Naïve Bayes. Dalam penelitian tersebut berhasil menghasilkan evaluasi dari algoritma Naïve Bayes yaitu akurasi sebesar 73,68%. Dalam hal ini terlihat bahwa algoritma Naïve Bayes dirasa masih memperoleh tingkat akurasi yang cukup rendah.

III. METODE PENELITIAN

A. Pre-processing data

Dalam tahap pre-processing data, dilakukan pembersihan teks dari karakter khusus, tanda baca, dan tokenisasi. Teks juga dikonversi

menjadi huruf kecil untuk konsistensi. Selanjutnya, dilakukan penghapusan kata-kata yang tidak relevan atau stop words, serta penerapan stemming atau lemmatization untuk mengubah kata-kata menjadi bentuk dasar.

B. Algoritma SVM

Support Vector Machine (SVM) adalah metode klasifikasi yang mencari hyperplane terbaik untuk memisahkan kelas-kelas dalam data dengan margin terbesar. Algoritma ini bekerja dengan cara memetakan data ke dalam ruang berdimensi tinggi menggunakan trik kernel dan menemukan hyperplane yang memaksimalkan margin antara kelas-kelas yang berbeda. Cara kerja SVM melibatkan penentuan support vectors, yaitu titik-titik data terdekat dengan hyperplane pemisah. SVM kemudian menggunakan fungsi kernel (seperti linear, polynomial, atau RBF) untuk menangani data yang tidak dapat dipisahkan secara linear. Model ini mengklasifikasikan data ke dalam kelas berdasarkan sisi hyperplane tempat data berada. Selanjutnya, kinerja model dievaluasi menggunakan metrik seperti akurasi, presisi, recall, atau F1-score.

C. Algoritma BI-LSTM

Bidirectional Long Short-Term Memory (BI-LSTM) merupakan sebuah pengembangan dari Jaringan Saraf Rekurens (RNN) yang diciptakan khusus untuk mengatasi keterbatasan yang dimiliki oleh RNN tradisional dalam memahami ketergantungan jangka panjang pada data berurutan. Secara umum, BI-LSTM terdiri dari dua lapisan LSTM yang beroperasi secara simultan dalam dua arah yang berlawanan, yaitu maju dan mundur, untuk menangkap informasi dari masa lalu dan masa depan dalam data sekuensial. Cara kerja BI-LSTM dimulai dengan pengambilan keputusan mengenai informasi mana yang penting untuk disimpan dan mana yang perlu diabaikan dari sel (cell) dalam kedua arah tersebut. Selanjutnya, dilakukan penentuan informasi baru yang relevan yang harus disimpan dan menggantikan informasi yang tidak relevan. Proses terakhir melibatkan penentuan output dengan bantuan sel yang telah difilter, di mana sel tersebut menjadi versi yang telah disaring melalui fungsi sigmoid dan tangen hiperbolik (tanh) yang sebelumnya telah diterapkan, baik dalam arah maju maupun mundur. Dengan demikian, BI-LSTM memungkinkan untuk

memahami dan menangkap pola data secara lebih baik dari RNN tradisional.

D. Dataset

Dataset mencakup deskripsi data yang digunakan dalam penelitian, termasuk sumber data, jumlah data, dan atribut-atribut yang ada dalam dataset. Data dapat berupa fakta, angka, nama, atau deskripsi. Dataset memiliki hubungan erat dengan kegiatan data mining, yang membantu peneliti dalam menganalisis data menjadi informasi yang koheren. Dataset yang digunakan dalam penelitian ini adalah dataset sentimen tweet Pilpres 2019. Dataset ini terdiri dari 1815 baris data dan 2 kolom data dengan variabel Sentiment dan Text Tweet.

E. Word2vec

Word2vec merupakan teknik embedding dengan kemampuan terbaik dalam merepresentasikan semantik dari kata-kata. Secara singkat, word2vec mengubah setiap kata menjadi vektor dan kata-kata yang memiliki kemiripan makna. Cara kerjanya melibatkan dua arsitektur utama, yaitu Continuous Bag of Words (CBOW) dan Skip-gram. Pada metode CBOW, model memprediksi sebuah kata berdasarkan konteks kata-kata sekitarnya, sementara pada metode Skip-gram, model memprediksi konteks sekitar sebuah kata tertentu. Word2Vec mempelajari representasi kata-kata dengan memetakan mereka ke dalam vektor berdimensi tinggi yang mempertahankan hubungan semantik antara kata-kata tersebut. Sebagai hasilnya, kata-kata yang memiliki makna serupa akan memiliki vektor yang berdekatan dalam ruang vektor. Proses pelatihan dilakukan melalui neural network yang dioptimalkan menggunakan algoritma backpropagation untuk meminimalkan kesalahan prediksi, sehingga model dapat menangkap hubungan semantik dan sintaksis antara kata-kata dalam teks.

F. Proses Training

Proses training pada algoritma SVM menggunakan kernel RBF dengan nilai gamma auto serta menggunakan kolom fitur tweet text dan kolom target Sentiment. Untuk algoritma BI-LSTM, proses training juga menggunakan kolom fitur dan target yang sama, dengan pembagian data uji sebesar 10% (test size 0.1) dan random state 0. Model dilatih dengan batch

size sebesar 64 dan jumlah epoch sebanyak 100, menggunakan optimizer Adam.

G. Evaluasi Model

Evaluasi model pada kedua algoritma, yaitu SVM dan BI-LSTM, dilakukan dengan menggunakan confusion matrix dan classification report. Evaluasi ini bertujuan untuk mengukur performa model dalam mengklasifikasikan data ke dalam dua kelas: 'class negatif' dan 'class positif'. Confusion matrix memberikan gambaran mengenai jumlah prediksi yang benar dan salah untuk setiap kelas, membantu dalam memahami kinerja model secara lebih mendetail.

Selain itu, classification report memberikan metrik evaluasi penting seperti precision, recall, dan F1-score untuk setiap kelas. Precision mengukur akurasi prediksi positif dari model, recall mengukur kemampuan model dalam menemukan semua instance positif, dan F1-score memberikan rata-rata harmonis dari precision dan recall. Dengan menggunakan kedua alat evaluasi ini, performa model dapat dianalisis secara komprehensif, memungkinkan peneliti untuk memahami sejauh mana model mampu mengklasifikasikan sentimen secara akurat dan efektif. Hasil evaluasi ini akan menjadi dasar untuk menentukan kelebihan dan kekurangan dari masing-masing algoritma dalam tugas analisis sentimen.

IV. GAMBARAN DESAIN SISTEM

A. Pengambilan Dataset

```

import pandas as pd
df = pd.read_excel('/content/sample_data/tweet1.xlsx')
df.head()

# Menyeimbangkan jumlah positif dan negatif sentimen
df_neg = df[df['sentimen'] == 0]
df_pos = df[df['sentimen'] == 1].head(596)
df = pd.concat([df_pos, df_neg], axis=0)
df.groupby('sentimen').describe()

```

	no	sentimen	tweet
0	0	0	Kata @prabowo Indonesia tidak dihargai bangsa ...
1	1	1	Batuan Langka, Tasbih Jokowi Hadiah dari Habib...
2	2	1	Di era Jokowi, ekonomi Indonesia semakin baik...
3	3	1	Bagi Sumatera Selatan, Asian Games berdampak p...
4	4	0	Negara kita ngutang buat bngun infrastruktur y...

	no	count	mean	std	min	25%	50%	75%	max
sentimen	0	596.0	925.731544	508.037094	0.0	511.5	953.0	1361.25	1805.0
	1	1219.0	897.841674	531.772698	1.0	427.5	894.0	1359.00	1814.0

Gambar 1. Pengambilan Dataset

Dari gambar diatas dapat diketahui dari dataset yang kami gunakan terdapat

596 data negatif (0) dan 1219 positif (1) yang nantinya akan diproses terlebih dahulu sebelum melalui tahapan modeling.

B. Preprocessing

1) Penghapusan Duplikasi Data

```
#Menghapus data duplikat
df = df.drop_duplicates()
```

Gambar 2. Penghapusan Duplikasi Data

Pada proses ini dilakukan penghapusan pada data duplikat yang bertujuan untuk menghemat ruang penyimpanan serta meningkatkan akurasi model karena data duplikat dapat menyebabkan bias dalam pelatihan model dan mengganggu proses learning sehingga mengganggu akurasi model.

2) Menyeimbangkan Jumlah Sentiment Negatif Positif



Gambar 3. Penyeimbang Sentiment

Langkah ini bertujuan untuk menyeimbangkan jumlah data positif dan negatif dalam analisis sentimen, memastikan distribusi yang proporsional antara kedua kategori, sehingga menghindari bias pada model terhadap salah satu sentimen. Kami menerapkan teknik undersampling, yaitu mengurangi sampel dari kelas yang dominan agar jumlahnya setara dengan kelas minoritas. Seperti yang terlihat pada ilustrasi, jumlah sentimen telah menjadi seimbang dengan total masing-masing 596.

3) Pembersihan Text

```
[ ] # set stop word
nltk.download('stopwords')
stop_words = stopwords.words('indonesian')

[ ] [nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

[ ] def preprocess_text(review):
    review = re.sub(r"http\S+", "", review)           # menghapus link website
    review = re.sub(r"\S*\d\S*", "", review).strip()   # proses menghapus nomor
    review = re.sub(r"[A-Za-z]+", "", review)          # proses menghapus selain huruf
    review = review.translate(str.maketrans("", "", string.punctuation)) # menghapus tanda baca
    review = review.lower()                           # proses case folding
    review = [word for word in review.split(" ") if not word in stop_words] # proses stop word
    review = " ".join(review)
    review.strip()
    return review

df['tweet_clean'] = df['tweet'].apply(lambda x: preprocess_text(x))
```

Gambar 4. Pembersihan Text

Pembersihan text ini terdiri dari beberapa bagian sebagai berikut :

- Menghapus tautan
Baris ini menggunakan fungsi re.sub untuk menghapus URL dari teks. Pola "http\S+" mencocokkan string yang dimulai dengan "http" diikuti oleh karakter non-spasi.
 - Menghapus angka
Baris ini menggunakan fungsi re.sub untuk menghapus kata-kata yang mengandung angka. Pola "\S*\d\S*" mencocokkan kata yang mengandung angka. Metode strip() menghapus spasi di awal dan akhir teks.
 - Menghapus karakter
Baris ini menggunakan fungsi re.sub untuk mengganti semua karakter non-huruf dengan spasi. Pola '[^A-Za-z]+' mencocokkan semua karakter yang bukan huruf (baik huruf besar maupun kecil).
 - Menghapus tanda baca
Baris ini menghapus semua tanda baca dari teks dengan menggunakan metode translate dan maketrans dari modul string.
 - Lowercasing
Baris ini mengubah semua huruf dalam teks menjadi huruf kecil dengan menggunakan metode lower().
 - Penghapusan kata pada stopword
Baris ini memecah teks menjadi kata-kata individu (menggunakan split("")) dan kemudian menghapus kata-kata yang ada dalam daftar stop_words dengan menggunakan list comprehension.

no	sentimen	tweet	tweet_clean
1	1	Batuan Langka, Tasbih Jokowi Hadiah dari Habib...	batuan langka tasbih jokowi hadiah habib lutfi...
2	2	Di era Jokowi, ekonomi Indonesia semakin baik...	era jokowi ekonomi indonesia jokowi laga jokow...
3	3	Bagi Sumatera Selatan, Asian Games berdampak p...	sumatra selatan asian games berdampak pd ekon...
5	5	Yg bisikin pak jokowi, cm mkin perputaran d...	yg bisikin jokowi cm mkin perputaran duit g...
6	6	Masa tenang msh apgoach aja...tp jokowi harga ...	tenang msh apgoach aja tp jokowi harga mat...
...
213	213	Maaf mas saya ga pake yg subsidi; saya pake yg...	maaf mas qk pake yg subsidi pake harga udah te...
214	214	Video dengan sudut pandang berbeda, mas @kow...	video sudut pandang berbeda jokowi prabowo vd...
216	216	Saya selalu berdoa dalam hal ini agar @prabow...	berdoa halab prabowo mjd presiden iku ekonomi ...
217	217	Presiden Jokowi - Raja Salman Sepakat Tingkat...	presiden jokowi raja salman sepakat tingkatkan...
219	219	Jgn * and a dpt harga 16rb karena itu tako aban...	jgn dpt harga tko abang

Gambar 5. Text Setelah Pembersihan

Berikut adalah data setelah dilakukan pembersihan.

4) Word2vec

Proses dibawah ini dilakukan untuk model deep learning pada BI-LSTM

- Tokenizing

Gambar 6. Tokenizing

Tokenizing bertujuan untuk memecah teks menjadi unit yang lebih kecil yang disebut **token**. Token ini biasanya merupakan kata, tetapi juga bisa berupa karakter atau sub-kata. Proses tokenizing memungkinkan model untuk menganalisis dan memproses teks secara sekuensial dan memahami konteks serta hubungan antar kata dalam kalimat. Dalam BI-LSTM, tokenizing sangat penting karena memungkinkan model untuk belajar dari data sekuensial baik dari masa lalu ke masa depan (forward pass) maupun sebaliknya (backward pass), sehingga meningkatkan kemampuan model dalam memahami konteks dua arah dalam teks.

- Word2vec

Model

```

from nltk.data import find
from gensim.models import KeyedVectors

# Download the word2vec sample model from nltk
nltk.download('word2vec_sample')

# Set the path to the model file
model_path = find("models/word2vec_sample/pruned.word2vec.txt")

# Load the word2vec model
model = KeyedVectors.load_word2vec_format(model_path, binary=False)

[nltk_data] Downloading package word2vec_sample to /root/nltk_data...
[nltk_data] Unzipping models/word2vec_sample.zip.

model_weights = model.vectors

vocab_size, embedding_size = model_weights.shape
print("Size Komas: %d - Dimensi Embedding: %d") % (vocab_size, embedding_size)

Size Komas: 43981 - Dimensi Embedding: 300

```

Gambar 6. Memodelkan Word2vec

Kami menggunakan model Word2Vec bahasa Indonesia dengan size kamus 43981 dan dimensi embedding 300.

- Vectorizer

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(ngram_range=(1, 2), stop_words=stop_words, min_df=20)
x = vectorizer.fit_transform(df['tweet_clean'])
y = df['sentiment']

/usr/local/lib/python3.10/dist-packages/sklearn/feature_extraction/text.py:409: UserWarning: warn()

text = vectorizer.transform(['prabowo jelek banget kayak badud'])
text = vectorizer.inverse_transform(text)
print(text)

[array(['banget', 'prabowo'], dtype='|S33')]
```

Gambar 7. Vectorizer

Proses ini digunakan untuk model SVM. Pada percobaan kali ini kami mengimpor kelas CountVectorizer dari modul sklearn.feature_extraction.text. CountVectorizer digunakan untuk mengubah koleksi dokumen teks menjadi matriks istilah-dokumen.

C) Training Model BI-LSTM

```

# construct model
BATCH_SIZE = 64

from tensorflow.keras.utils import plot_model # Import from tensorflow.keras
model = Sequential()
model.add(Embedding(len(tokenizer.index_word)+1, 64))
model.add(Bidirectional(LSTM(100, dropout=0, recurrent_dropout=0)))
model.add(Dense(100, activation="relu"))
model.add(Dense(1, activation="sigmoid"))

model.compile("adam", "binary_crossentropy", metrics=["accuracy"])
plot_model(model, to_file="model_plot.png", show_shapes=True, show_layer_names=True)

Model: "sequential"
-----  

Layer (type)          Output Shape       Param #
=====  

embedding (Embedding) (None, None, 64)    353536  

bidirectional (Bidirection (None, 200)    112000
al)  

dense (Dense)         (None, 128)        25778  

dense_1 (Dense)       (None, 1)          129  

total params: 111081 (1.05 MB)
Trainable params: 511393 (1.95 MB)
Non-trainable params: 0 (0.00 Byte)
-----  

from sklearn.metrics import classification_report

# Prediksi kelas dari data uji
y_pred = model.predict(X_test_seq_padded)
y_pred = (y_pred > 0.5).astype(int) # Ubah probabilitas menjadi kelas biner (0 atau 1)

# Buat laporan klasifikasi
report = classification_report(y_test, y_pred, target_names=['class negatif', 'class positif'])

# Print laporan klasifikasi
print("Classification Report:")
print(report)

15/15 [=====] - 0s/3ms/stop
Classification Report:
precision    recall   f1-score   support
class negatif  0.60    0.64    0.62    161
class positif  0.83    0.81    0.82    313

accuracy      0.71    0.72    0.72    454
macro avg     0.71    0.72    0.72    454
weighted avg  0.76    0.75    0.76    454

```

Gambar 8. Memodelkan BI-LSTM

D) Training Model SVM

```
from sklearn import svm
#The 'kernel' parameter of SVC must be a str among ('linear', 'precomputed', 'sigmoid', 'poly', 'rbf')
clf = svm.SVC(kernel='linear', gamma='auto')
clf.fit(x_train, y_train)

# SVC(gamma='auto', kernel='linear')

ypredict = clf.predict(x_test)
ypredict [0:5]

array([1, 1, 1, 1, 1])
```

Gambar 9. Training Model SVM

Setelah melakukan pengujian kernel terbaik yang kami gunakan adalah linear

```

from sklearn.metrics import classification_report, confusion_matrix
#Printing accuracy,confusion matrix,Precision and Recall
print("Train set Accuracy: ", metrics.accuracy_score(y_train, clf.predict(x_train)))
print("Test set Accuracy: ", metrics.accuracy_score(y_test, ypredict))

print("\nAccuracy of the classifier is",metrics.accuracy_score(y_test,ypredict))
print("\nConfusion matrix is",metrics.confusion_matrix(y_test,ypredict))
print("\nClassification report is",metrics.classification_report(y_test,ypredict))
print("\nThe value of Precision",metrics.precision_score(y_test,ypredict))
print("\nThe value of Recall",metrics.recall_score(y_test,ypredict))

Train set Accuracy:  0.7954684629516228
Test set Accuracy:  0.6923076923076923

Accuracy of the classifier is 0.6923076923076923

Confusion matrix is:
[[ 19  38]
 [ 18 107]]

classification report is:
              precision    recall   f1-score   support
          0       0.51      0.33      0.40      57
          1       0.74      0.86      0.79     125

   accuracy                           0.69      182
  macro avg       0.63      0.59      0.60      182
weighted avg       0.67      0.69      0.67      182

The value of Precision 0.7379310344827587
The value of Recall 0.856

```

Gambar 10. Hasil Pengujian

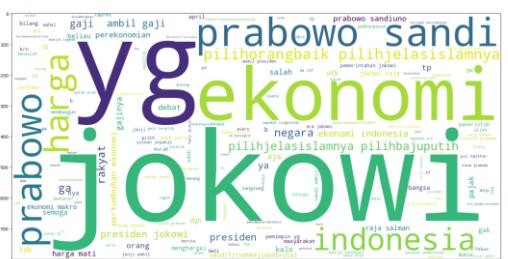
E) Implementasi dan Pengujian

1) WordClouds

- Sentimen Negatif

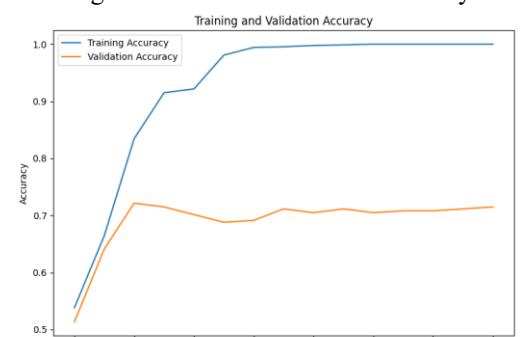


- Sentimen Positif

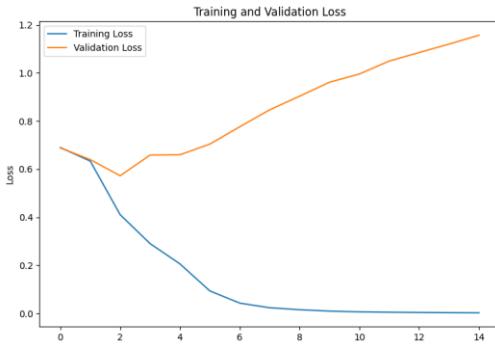


2) BI-LSTM

- #### • Training and Validation Accuracy



- Training and Validation Loss



- Accuracy dan ROC AUC Score

```
# Prediksi kelas dari data uji
y_pred = model.predict(X_test_seq_padded)
y_pred_binary = (y_pred > 0.5).astype(int) # Ubah probabilitas menjadi kelas biner (0 atau 1)

# Hitung nilai akurasi
accuracy = accuracy_score(y_test, y_pred_binary)

# Cetak nilai akurasi
print("Accuracy:", accuracy)

10/10 [=====] - 1s 85ms/step
Accuracy: 0.71476510671141

# Prediksi probabilitas dari data uji
y_pred = model.predict(X_test_seq_padded)

# Hitung nilai ROC AUC score
roc_auc = roc_auc_score(y_test, y_pred)

# Cetak nilai ROC AUC score
print("ROC AUC Score:", roc_auc)

10/10 [=====] - 0s 45ms/step
ROC AUC Score: 0.7738963963963963
```

3) SVM

- Confusion Matrix

		Confusion Matrix	
		negative	positive
True	negative	38	25
	positive	24	33
	Predicted	negative	positive

V. KESIMPULAN

Berdasarkan penelitian yang sudah kami lakukan, dapat ditarik kesimpulan bahwa analisis sentimen menggunakan Metode Support Vector Machines dan Long-Short Term Memory dapat diimplementasikan dalam klasifikasi sentimen terhadap calon presiden 2019. Dari hasil pengujian pada metode BI-LSTM memiliki tingkat akurasi sebesar 0.714, sedangkan pada metode SVM memiliki tingkat akurasi sebesar 0.591. Maka dapat kami simpulkan dari kedua metode, metode Bi-LSTM mempunyai performa measure atau akurasi cukup tinggi dibandingkan dengan metode SVM.

Dari analisis yang dilakukan, terlihat bahwa metode Bi-LSTM mampu menangkap kompleksitas dan nuansa dalam teks tweet dengan lebih baik dibandingkan SVM, yang cenderung kurang efektif dalam memproses data teks yang tidak terstruktur dan memiliki konteks yang mendalam. Keunggulan Bi-LSTM ini dapat dikaitkan dengan kemampuannya untuk mempertimbangkan urutan kata dan konteks secara lebih holistik melalui pemrosesan dua arah, sehingga mampu memberikan representasi yang lebih kaya dari teks.

DAFTAR PUSTAKA

- Suryati, E., Styawati, S., & Aldino, A. A. (2023). Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Dan Sistem Informasi*, 4(1), 96-106.
- Husada, H. C., & Paramita, A. S. (2021). Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM). *Teknika*, 10(1), 18-26.
- Sabily, A. F., Adikara, P. P., & Fauzi, M. A. (2019). Analisis sentimen pemilihan presiden 2019 pada twitter menggunakan metode maximum entropy. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(5), 4204-4209.
- Kamarula, M. R. F., & Rochmawati, N. (2022). Perbandingan CNN dan Bi-LSTM pada Analisis Sentimen dan Emosi Masyarakat Indonesia Di Media Sosial Twitter Selama Pandemik Covid-19 yang Menggunakan Metode Word2vec. *Journal of Informatics and Computer Science (JINACS)*, 219-228.
- Farasalsabila, F., Utami, E., & Hanafi, H. (2024). Deteksi Cyberbullying Menggunakan BERT dan Bi-LSTM. *Jurnal Teknologi*, 17(1), 1-6.