



## Perbandingan Akurasi Analisis Sentimen *Tweet* terhadap Pemerintah Provinsi DKI Jakarta di Masa Pandemi

Ragil Dimas Himawan<sup>#1</sup>, Eliyani<sup>\*2</sup>

<sup>#</sup>Program Studi Informatika Fakultas Ilmu Komputer, Universitas Mercu Buana  
Jl. Meruya Selatan No. 1, Kembangan, Jakarta 11650

<sup>1</sup>41516110088@student.mercubuana.ac.id

<sup>2</sup>elijani@mercubuana.ac.id

**Abstrak**— Pemerintah memanfaatkan media sosial seperti twitter sebagai salah satu kanal interaksi dengan masyarakat. Informasi hasil interaksi tersebut sebagai umpan balik untuk mengetahui opini masyarakat terhadap kebijakan publik. Analisis sentimen *tweet* dari masyarakat dapat dijadikan sebagai salah satu parameter penunjang bagi pemerintah dalam mengevaluasi kebijakan dan pengambilan keputusan mendatang. Penelitian ini bertujuan untuk mengetahui sentimen data *tweet* masyarakat terhadap akun twitter resmi Pemerintah Provinsi DKI Jakarta di masa pandemi COVID-19. Data yang diperoleh sebanyak 14208 baris dengan *query* pada *tweet* yang mengandung kata atau menyebut *username @dkijakarta*, dimana akan dikelompokkan berdasarkan kelas sentimen yaitu, negatif, netral, dan positif dengan menggunakan *TF-IDF Vectorizer* untuk pembobotan kata dan klasifikasi menggunakan beberapa metode yaitu, *random forest classifier* dengan hasil akurasi sebesar 75,81%, algoritma *naive bayes* dengan hasil akurasi 75,22%, dan algoritma *support vector machine* 77,58%. Dilakukan proses analisis sentimen pada *tweet* dengan presentase hasil negatif, netral, dan positif masing-masing yaitu, 8,8%, 83,6%, 7,6%.

**Kata kunci**— Analisis Sentimen, Twitter, Text Mining, TF-IDF, Klasifikasi, Support Vector Machine, Naive Bayes, Random Forest Classifier

### I. PENDAHULUAN

Perkembangan teknologi internet telah membuat penyebaran informasi meningkat secara signifikan. Salah satu yang mendukung penyebaran informasi tersebut adalah media sosial. Pengguna media sosial bukan hanya sebagai konsumen informasi, tetapi juga sebagai penghasil informasi[1]. Salah satu media sosial dengan pengguna terbanyak adalah twitter. Indonesia memiliki jumlah pengguna aktif twitter yang tinggi[2]. Hal ini berdampak pula pada tingginya jumlah data *tweet* yang dihasilkan. Berbagai peristiwa atau topik terkini menjadi pemicu bagi setiap pengguna untuk melakukan *tweet*.

Berbagai instansi pemerintahan sudah memiliki akun twitter resmi yang digunakan sebagai kanal interaksi antara masyarakat dan pihak pemerintah[3]. Salah satunya adalah

Pemerintah Provinsi DKI Jakarta dengan *username @dkijakarta*. Melalui twitter, masyarakat dapat mengutarakan opini mengenai kebijakan pemerintah, mengajukan pertanyaan mengenai pelayanan publik, ataupun sekadar berkomentar untuk bersosialisasi antar masyarakat.

*Coronavirus disease* 2019 atau *COVID-19* ditetapkan secara resmi oleh Organisasi Kesehatan Dunia sebagai pandemi global pada 11 Maret 2020[4], berbagai kebijakan diambil oleh Pemerintah Provinsi DKI seperti Pembatasan Sosial Berskala Besar atau PSBB yang mulai berlaku pada 10 April 2020. PSBB mencakup hal terkait kegiatan perekonomian, keagamaan, sosial, budaya dan pendidikan di Jakarta[5]. PSBB secara tidak langsung berpengaruh terhadap perilaku masyarakat untuk mengurangi interaksi secara langsung antar individu. Twitter merupakan media sosial yang banyak digunakan masyarakat saat pandemi.

Sumber data penelitian ini berasal dari *tweet* masyarakat terhadap akun twitter Pemerintah Provinsi DKI Jakarta di masa pandemi untuk dilakukan klasifikasi dalam menentukan sentimen. Pada penelitian terkait sebelumnya[6], akurasi metode *Naive Bayes* lebih baik dibandingkan dengan *k-Nearest Neighbor(k-NN)*. Penelitian lainnya[7] menggunakan *support vector machine* pada dua dataset dengan topik berbeda yaitu *self-driving car* dan produk *apple*. Akurasi yang dihasilkan masing-masing sebesar 59,91% dan 71,2%. Penelitian yang lain[8] didapatkan skenario terbaik dengan teknik klasifikasi *support vector machine* menggunakan pembobotan kata *TF-IDF* dan *stemming*.

Pada penelitian ini data diambil dengan teknik *crawling* menggunakan API Twitter pada rentang waktu 9 April 2020 sampai 15 April 2020. Metode yang digunakan yaitu *support vector machine*, *naive bayes*, dan *random forest classifier* dengan *TF-IDF Vectorizer* dan proses *stemming* bahasa Indonesia. Metode dengan model terbaik akan digunakan untuk memprediksi sentimen pada data yang kosong.

II. LANDASAN TEORI

A. Analisis Sentimen

Analisis sentimen adalah metode komputasional untuk mengekstraksi dan menganalisis sentimen pada suatu entitas dan atribut yang dimiliki[9][10]. Pada umumnya menggunakan metode dengan pendekatan *machine learning* berbasis teks.

B. Text Mining

*Text mining* merupakan proses penemuan pengetahuan menggunakan *Natural Language Processing* (NLP) dengan cara menggali informasi dari sebuah data berformat teks[11]. Data teks bisa berupa data yang terstruktur seperti data dalam *database* maupun data yang tidak terstruktur seperti kumpulan dari dokumen teks.

C. TF-IDF

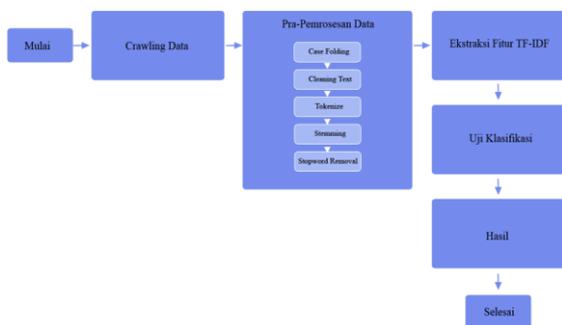
TF-IDF adalah metode pembobotan kata dengan mengekstraksi ciri dari suatu teks[12]. Terdiri dari dua aspek: (1) *term frequency*(TF), frekuensi sebuah *term* muncul dalam sebuah dokumen; (2) *inversed document frequency* (IDF), mengukur seberapa penting suatu *term*[13]. Dalam TF, semua istilah dianggap sama pentingnya. IDF adalah ukuran besarnya kepentingan *term* yang diimbangi dengan frekuensi suatu istilah yang muncul dalam *dataset*.

D. Klasifikasi

Klasifikasi merupakan sebuah teknik yang digunakan untuk memasukan objek ke dalam kelas-kelas sesuai dengan karakteristik kelas yang telah didefinisikan sebelumnya[14]. Proses klasifikasi berdasarkan pola yang didapatkan dari data historis yang lalu dan digunakan sebagai klasifikasi nilai yang diprediksi di masa mendatang.

III. METODE PENELITIAN

Pada penelitian ini terdiri dari beberapa tahapan meliputi crawling data, pra-pemrosesan data, ekstraksi fitur TF-IDF, uji klasifikasi dan hasil seperti pada Gambar 1.



Gambar 1. Diagram kerja

A. Crawling Data

Metode pengumpulan data dilakukan dengan teknik *crawling* memanfaatkan *API key* twitter. Data diambil dengan kata kunci @dkijakarta dan difilter hanya yang

berbahasa Indonesia dan bukan merupakan hasil dari *retweet*.

```

30 consumer_key = 'HnLHtHoytJJV1DNZnd2b8TsJS'
31 consumer_secret = 'WzmJjgI041U3HpRmyjPChW8VMXTAjUzaiXfCo3LM2s7aoD3wF'
32 access_token = '278989147-xYDFVdGs4DtjS2UiRdLMhBJa04qh8DwkvyABigFt'
33 access_secret = 'kDxz10Dgo6FsTha1fgBOZBYK5BbEqmdrOjcs5p7GPGPv9Z'
34 auth = OAuthHandler(consumer_key, consumer_secret)
35 auth.set_access_token(access_token, access_secret)
36 # load the twitter API via tweepy
37 return tweepy.API(auth)
    
```

Gambar. 2 API key twitter

```

10 tweet_list = []
11 text_query = '@DKIJakarta -filter:retweets|'
12 count = 100
13 try:
14 # Pulling tweet from query
15 for tweet in api.search(q=text_query, count=count, lang="id"):
16 # Adding to list that contains all tweets
17 tweets.append((tweet.created_at, tweet.id, tweet.text))
    
```

Gambar. 3 Query menggunakan fungsi *api.search*

B. Pra-Pemrosesan Data

Tahapan pra-pemrosesan data adalah tahap pembersihan data dan mentransformasikan data teks menjadi lebih terstruktur. Tahapan ini diperlukan untuk meningkatkan kualitas data dengan menghapus data yang tidak diinginkan dari data asli[15]. Ada beberapa tahapan yang digunakan pada pra-pemrosesan data teks, yaitu :

- 1) *Case Folding*: Mentransformasi semua huruf yang ada dalam dokumen menjadi huruf kecil.
- 2) *Cleaning text*: Penghapusan karakter, simbol, *username* (@username), URL dan tanda baca yang tidak diperlukan.
- 3) *Tokenize*: Pemecahan dokumen teks atau kalimat-kalimat yang ada ke dalam potongan-potongan kata yang disebut token.
- 4) *Stemming*: Proses *stemming* adalah proses transformasi sebuah kata menjadi kata dasar (root). Untuk *stemming* dalam bahasa Indonesia menggunakan *library python* sastrawi. Sastrawi merupakan *library* yang dapat mengubah kata berimbuhan dalam bahasa Indonesia menjadi bentuk kata dasar.
- 5) *Stopword removal*: penghapusan kata yang terdapat pada *stopword list* yang berisi kata-kata tidak penting atau tidak memiliki makna seperti kata hubung ataupun kata depan.

C. TF-IDF

TF-IDF dilakukan untuk pembobotan kata dari setiap data *tweet*. Rumus dari TF-IDF adalah

$$TF = \frac{\text{Frekuensi term dalam satu dokumen}}{\text{Total kata dalam dalam satu dokumen}} \quad (1)$$

$$IDF = \log \frac{\text{Total dokumen} + 1}{\text{Frekuensi dokumen mengandung term}} \quad (2)$$

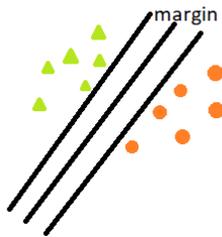
$$TF - IDF = TF \times IDF \quad (3)$$

Pada penelitian ini menggunakan *library scikit-learn* dalam implementasi TF-IDF untuk mengubah data menjadi bentuk vektor.

D. Klasifikasi

Ada beberapa metode klasifikasi yang digunakan:

1) *Support Vector Machine (SVM)*: Cara kerja SVM adalah berusaha menemukan *hyperplane* dengan jarak antar kelas[16].



Gambar. 4 *Hyperplane* pada SVM

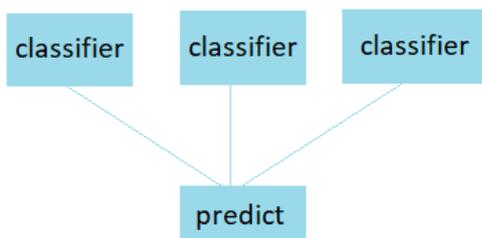
Implementasi metode SVM menggunakan salah satu class yang tersedia pada *library scikit-learn* yaitu, *LinearSVC*.

2) *Naïve Bayes*: Metode ini menggunakan hitungan probabilitas bersyarat atau biasa dikenal dengan Teorema Bayes. Pada *Text mining*, jenis yang cocok dipakai adalah *Multinomial Naive Bayes*[17].

```
>>> import numpy as np
>>> rng = np.random.RandomState(1)
>>> X = rng.randint(5, size=(6, 100))
>>> y = np.array([1, 2, 3, 4, 5, 6])
>>> from sklearn.naive_bayes import MultinomialNB
>>> clf = MultinomialNB()
>>> clf.fit(X, y)
MultinomialNB()
>>> print(clf.predict(X[2:3]))
[3]
```

Gambar. 5 *Class multinomialNB* pada *scikit-learn*

3) *Random Forest Classifier (RF)*: RF adalah algoritma *ensemble* yang merupakan kombinasi dari pohon keputusan (*classifier*)[18]. RF menggunakan sampling secara acak dengan distribusi yang seimbang.



Gambar. 6 *Random forest*

IV. HASIL DAN PEMBAHASAN

Data yang berhasil diambil dengan teknik *crawling* menggunakan API Twitter pada rentang waktu 9 April 2020 sampai 15 April 2020. *Query* yang digunakan adalah @dkijakarta kemudian difilter berdasarkan Bahasa Indonesia dan *tweet* yang bukan merupakan hasil dari *retweet* menghasilkan 14208 baris data.

Tanggal	text	screen_name
4/15/2020	@DKIJakarta Pemimpin kw itu ketika urusan formal di omongin via MEDSOS!!! Ketularan berobat ONLINE	BangDama
4/15/2020	yg masih menyuruh datang ke tempat kerja nya, padahal pemprov DKI... https://t.co/HVAwkVFYS7	Hern22003036
4/15/2020	@DKIJakarta KI di daerah cakung , boleh buka tapi dibawa pulang . Jd gak ada bangku buat mkn	Dodi76650439
4/15/2020	tindakan kepada warga yang masih nongkrong &gt;5 orang di daerah Jal... https://t.co/Ud0ywMJ18V	retnowiryawan
4/15/2020	jamet, nyaut ngape lo....	jajaka666

Gambar. 7 Hasil *crawling*

Setelah mendapatkan data hasil *crawling* dan dijadikan sebuah dataset, tahap selanjutnya adalah proses pelabelan pada sebagian data untuk membagi kelas sentimen menjadi tiga kelas, yaitu negatif, netral, dan positif. Pelabelan dilakukan secara manual untuk kemudian dibagi menjadi dua bagian, data latihan dan data uji.

tweet	username	sentimen
@fadjroeL @DKIJakarta Bahu membahu...kerja sama yg top dech...	cueng81	1
@DKIJakarta Semoga dengan ada kebijakan ini jadi makin berkurang dan memutus rantai virus #CegahBarengCOVID19	Agustina_DJ7	1
Min @DKIJakarta pengambilan sembako pangan murah untuk guru KKI selama PSBB dan pandemi corona bagaimana ya rules nya?	Toxy_91	0
@Ghazi10415477 @DKIJakarta @kominfotikJU Patrolinya berbondong2..poto2 video2..ya gak efektif jadinya..	digitalmbul	-1
@KemenkesRI @DKIJakarta Selama gak tegas terkait kerumunan, ya percuma aja ada PSBB.	amirsyahoke	-1

Gambar. 8 Data setelah diberi label kelas sentimen

Pada Gambar 8. pelabelan kelas sentimen menggunakan angka untuk menentukan nilai kelasnya yaitu, (-1), (0), dan (1), yang masing-masing mewakili kelas negatif, netral, dan positif. Tahapan selanjutnya adalah pra-pemrosesan data.

```
# Preprocessor
def my_preprocessor(mytext):
    #Convert to Lower case
    mytext = mytext.lower()
    #Remove additional code
    mytext = mytext.replace('\x20\x20\x20', '')
    #Convert www.* or https:// to URL
    mytext = re.sub('((www|https?)|https?://[^\s]+)', 'URL', mytext)
    #Convert @username to AT_USER
    mytext = re.sub('@[^\s]+', 'ATUSER', mytext)
    #Remove additional white spaces
    mytext = re.sub('[\s]+', ' ', mytext)
    #Replace #word with word
    mytext = re.sub(r'#([^\s]+)', r'\1', mytext)
    #Menghapus angka dari teks
    mytext = re.sub(r'[0-9]+', '', mytext)
    #Menghapus tanda baca
    mytext = mytext.translate(str.maketrans(string.punctuation, '' * len(string.punctuation)))
    return mytext
```

Gambar. 9 *Cleaning text*

Pada Gambar 9 merupakan proses mentransformasi semua huruf menjadi huruf kecil, menghapus bentuk *URL link*, dan menghapus karakter serta tanda baca yang tidak diperlukan.

```
# Create Sastrawi stemmer
stemmer = StemmerFactory().create_stemmer()

# Tokenizer
def my_tokenizer(mytext):
    words = word_tokenize(mytext)
    tokens=[]
    for w in words:
        #add tokens
        if w not in ['ATUSER','URL'] and len(w) > 3:
            w = stemmer.stem(w)
            tokens.append(w.lower())
    return tokens
```

Gambar. 10 Proses tokenize dan stemming

Pada Gambar 10 dilakukan proses *tokenize* atau pemecahan data *tweet* menjadi bentuk token dengan *library* NLTK. Dalam proses tersebut juga dilakukan proses *stemming* menggunakan *library sastrawi* yang berfungsi untuk mengubah kata berimbuhan ke bentuk dasarnya atau yang biasa disebut *root*.

```
# Create Stopword
f = open("Kamus/Stopword.txt", "r")
my_stop_words = f.readlines()
f.close()
my_stop_words = my_stop_words.split()

print(my_stop_words)

['ada', 'adalah', 'adanya', 'adapun', 'agak', 'agakny', 'agar', 'akan', 'akank', 'akhir', 'akhir', 'akhirnya', 'aku', 'akulah', 'amat', 'amatlah', 'anda', 'andalah', 'antar', 'antara', 'antaranya', 'anu', 'apa', 'apaan', 'apabila', 'apakah', 'apalagi', 'apatah', 'artinya', 'asal', 'asalkan', 'atas', 'atau', 'ataukah', 'ataupun', 'awal', 'awalnya', 'bagai', 'bagaikan', 'bagaimana', 'bagaima
```

Gambar. 11 Stopword list

*Stopword* yang terdapat pada Gambar 11 merupakan file yang berisi kumpulan kata yang tidak penting atau kata yang tidak memiliki makna. Penghapusan *stopword* berguna untuk mengurangi kata yang tidak perlu diproses pada tahap *feature extraction* yang dapat mempengaruhi *performance*.

Dengan menggunakan *library scikit-learn*, class *TfidfVectorizer* dapat menghitung nilai TF-IDF sekaligus melakukan *pre-processing*.

```
# Modelling
word_count = vectorizer_count.fit_transform(data)
vectorizer_tfidf = TfidfVectorizer(preprocessor=my_preprocessor,
                                  tokenizer=my_tokenizer,
                                  stop_words=my_stop_words,
                                  min_df=5,
                                  max_df=0.85)
word_tfidf = vectorizer_tfidf.fit_transform(data)
```

Gambar. 12 Proses TF-IDF Vectorizer

Pada Gambar 12 adalah proses TF-IDF *vectorizer* yang merupakan proses transformasi data teks ke dalam bentuk vektor.

```
dimensions = word_count.get_shape()
print("total document = %s" % dimensions[0])
print("total feature = %s" % dimensions[1])
```

total document = 14208  
total feature = 2087

Gambar. 13 Hasil total ekstraksi fitur

Hasil total dari ekstraksi fitur yang dihasilkan dari proses TF-IDF *vectorizer* adalah 2087 fitur dan 14208 dokumen. Pada penelitian ini yang dimaksud dokumen adalah data *tweet*.

Uji klasifikasi dengan tiga metode yaitu, *Random Forset Classifier*, *Linear SVM*, dan *Multinomial Naïve Bayes*, seperti pada Gambar 14. Hasil uji akurasi dapat dilihat pada Gambar 15. Uji validasi menggunakan 10 *cross validation* untuk data latih dan data uji.

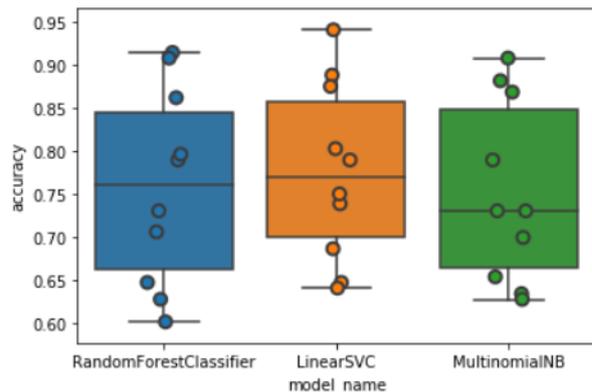
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import cross_val_score
from sklearn import metrics

names = ["Random Forest Classifier", "Linear SVM", "NB"]
models = [
    RandomForestClassifier(),
    LinearSVC(C=0.4),
    MultinomialNB(),
]
CV = 10
cv_df = pd.DataFrame(index=range(CV * len(models)))
entries = []
for model in models:
    model_name = model.__class__.__name__
    accuracies = cross_val_score(model, X, Y, scoring='accuracy', cv=CV)
    for fold_idx, accuracy in enumerate(accuracies):
        entries.append((model_name, fold_idx, accuracy))
cv_df = pd.DataFrame(entries, columns=['model_name', 'fold_idx', 'accuracy'])

import seaborn as sns
sns.boxplot(x='model_name', y='accuracy', data=cv_df)
sns.stripplot(x='model_name', y='accuracy', data=cv_df,
              size=8, jitter=True, edgecolor="gray", linewidth=2)
plt.show()

for i in range(len(names)):
    print(f"Accuracy {names[i]} : ", metrics.accuracy_score(y_test, y_pred))
    print(metrics.classification_report(y_test, y_pred, Y.unique()))
```

Gambar. 14 Uji model klasifikasi



Gambar. 15 Perbandingan akurasi

Didapatkan hasil akurasi pada masing-masing model klasifikasi yaitu, *Linear SVM* sebesar 77,58%, *Multinomial Naive Bayes* sebesar 75,22%, dan *Random Forset Classifier* sebesar 75,81%, seperti pada Gambar 16.

```

model_name
LinearSVC           0.775847
MultinomialNB      0.752227
RandomForestClassifier 0.758170
Name: accuracy, dtype: float64
    
```

Gambar. 16 Nilai akurasi masing-masing metode

Dengan tahapan dan proses yang sama terdapat perbedaan hasil akurasi dari setiap metode untuk klasifikasi data *tweet*. Hal ini dikarenakan karakteristik algoritma dari setiap metode *text classification*. Pada penelitian ini pendekatan dengan metode *linear SVM* mendapatkan hasil yang terbaik.

Tahap selanjutnya adalah memprediksi nilai sentimen pada data yang masih kosong menggunakan metode *Linear SVM*, seperti terlihat pada Gambar 17.

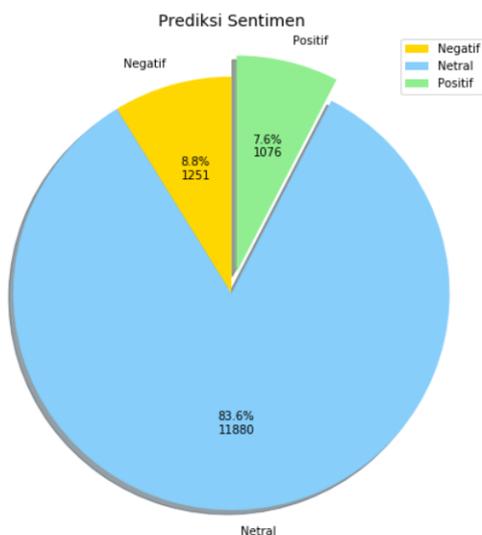
```

# Memprediksi menggunakan Linear SVM
clf = LinearSVC(C=0.4).fit(X_train,y_train)
test = word_tfidf[Data.index[Data['sentimen'].isna()]]
Data['sentimen'][Data['sentimen'].isna()] = clf.predict(test)

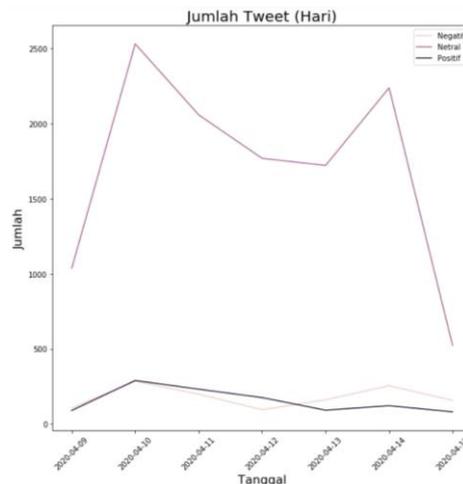
# Checking data
Data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 14208 entries, 0 to 14207
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   tanggal     14208 non-null  datetime64[ns]
1   tweet       14208 non-null  object
2   sentimen    14208 non-null  float64
dtypes: datetime64[ns](1), float64(1), object(1)
memory usage: 444.0+ KB
    
```

Gambar 17. Prediksi nilai sentiment



Gambar. 18 Presentase nilai sentiment



Gambar. 19 Jumlah *tweet* per hari

Pada Gambar 18 dan Gambar 19 dapat dilihat hasil prediksi nilai sentimen menggunakan metode *linear SVM* pada data yang kosong. Kelas sentimen dengan label netral memiliki presentase paling besar, 83,6%. Dua kelas lainnya yaitu, negatif dan positif masing-masing memiliki presentase sebesar 8,8% dan 7,6%.

Dari Gambar 19 terlihat, jumlah *tweet* harian mengalami kenaikan yang signifikan pada 10 April 2020 yang merupakan hari pertama pemberlakuan kebijakan Pembatasan Sosial Berskala Besar oleh Pemerintah terkait di Jakarta.

## V. KESIMPULAN DAN SARAN

Metode *Linear SVM*, *Multinomial Naive Bayes*, dan *Random Forrest Classifier* memiliki tingkat akurasi yang berbeda dalam mengklasifikasi data *tweet* masyarakat terhadap Pemerintah DKI Jakarta. *Linear SVM* memiliki akurasi terbaik dengan hasil 77,58%, *Random Forrest Classifier* dengan hasil 75,81%, dan *Multinomial Naive Bayes* sebesar 75,22%.

Hasil prediksi nilai kelas sentiment pada data yang kosong menggunakan algoritma *Linear SVM* menghasilkan prediksi netral sebesar 83,6%, negative 8,8% dan positif 7,6%.

Untuk penelitian yang akan datang penulis menyarankan untuk menggunakan lebih banyak data latih dan data uji agar mendapatkan nilai akurasi yang lebih tinggi. Selain itu, dapat dilakukan dengan metode atau algoritma yang dimodifikasi menjadi lebih baik.

## REFERENSI

- [1] A. P. Sitorus, H. Murfi, S. Nurrohmah, and A. Akbar, "Sensing Trending Topics in Twitter for Greater Jakarta Area," vol. 7, no. 1, pp. 330–336, 2017, doi: 10.11591/ijece.v7i1.pp330-336.
- [2] N. Y. A. Faradhillah, R. P. Kusumawardani, and I. Hafidz, "Eksperimen Sistem Klasifikasi Analisa Sentimen Twitter pada Akun Resmi Pemerintah Kota Surabaya Berbasis Pembelajaran Mesin (Experiments on Sentiment Classification System for Tweets of the Official Account of the City Government of Surabaya based on Mach)," *Pros. Semin. Nas. Sist. Inf. Indones. 2016*, pp. 15–24, 2016.
- [3] M. D. Anggreani, E. P. Purnomo, and A. N. Kasiwi, "RUANG PUBLIK VIRTUAL SEBAGAI PINTU KOMUNIKASI

- GOVERNMENT TO CITIZEN (Studi Kasus: Perbandingan Media Sosial Pemerintah Kota Yogyakarta dan Surabaya),” *J. MODERAT*, vol. 6, no. 1, pp. 203–220, 2020.
- [4] D. Cucinotta and M. Vanelli, “WHO declares COVID-19 a pandemic,” *Acta Biomed.*, vol. 91, no. 1, pp. 157–160, 2020, doi: 10.23750/abm.v91i1.9397.
- [5] W. Mas’udi and P. S. Winarti, *Tata Kelola Penanganan COVID-19 di Indonesia: Kajian Awal*. Jogjakarta: Gadjah Mada University Press Dilarang, 2020.
- [6] A. M. Pudjajana and D. Manongga, “SENTIMEN ANALISIS TWEET PORNOGRAFI KAUM HOMOSEKSUAL INDONESIA DI TWITTER DENGAN NAIVE BAYES,” vol. 9, no. 1, pp. 313–318, 2018.
- [7] M. Ahmad, S. Aftab, and I. Ali, “Sentiment Analysis of Tweets using SVM,” *Int. J. Comput. Appl.*, vol. 177, no. 5, pp. 25–29, 2017, doi: 10.5120/ijca2017915758.
- [8] K. M. Alomari, H. M. Elsharif, and K. Shaalan, “Arabic Tweets Sentimental Analysis Using Machine Learning,” vol. 1, no. June, pp. 602–610, 2017, doi: 10.1007/978-3-319-60042-0.
- [9] M. Soleymani, D. Garcia, B. Jou, B. Schuller, S. F. Chang, and M. Pantic, “A survey of multimodal sentiment analysis,” *Image Vis. Comput.*, vol. 65, pp. 3–14, 2017, doi: 10.1016/j.imavis.2017.08.003.
- [10] U. Rofiqoh, R. S. Perdana, and M. A. Fauzi, “Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode Support Vector Machine dan Lexion Based Feature,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 1, no. 12, pp. 1725–1732, 2017.
- [11] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safei, E. D. Trippe, J. B. Gutierrez, K. Kochut, “A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques,” 2017.
- [12] P. H. Saputro, M. Aristin, and Dy. L. Tyas, “Klasifikasi Lagu Daerah Indonesia Berdasarkan Lirik Menggunakan Metode TF-IDF dan Naive Bayes,” *J. Teknoloi Inform. dan Terap.*, vol. 4, no. 1, pp. 45–50, 2017.
- [13] Z. Zhu, J. Liang, D. Li, H. Yu, and G. Liu, “Hot Topic Detection Based on a Refined TF-IDF Algorithm,” *IEEE Access*, vol. 7, pp. 26996–27007, 2019, doi: 10.1109/ACCESS.2019.2893980.
- [14] S. Dwiasnati and Y. Devianto, “Optimasi Prediksi Keputusan Calon Nasabah Potensial Berbasis Particle Swarm Optimization,” *Optimasi Prediksi Keputusan Calon Nasabah Potensial Berbas. Part. Swa*, vol. 6, no. 2, pp. 286–292, 2019.
- [15] M. Sadikin and F. Alfiandi, “Comparative Study of Classification Method on Customer Candidate Data to Predict its Potential Risk,” *Int. J. Electr. Comput. Eng.*, vol. 8, no. 6, p. 4763, 2018, doi: 10.11591/ijece.v8i6.pp4763-4771.
- [16] D. Suyanto, *Data Mining Untuk Klasifikasi dan Klasterisasi Data*, Edisi Revi. Bandung: Penerbit Informatika, 2019.
- [17] L. Jiang, S. Wang, C. Li, and L. Zhang, “Structure extended multinomial naive Bayes,” *Inf. Sci. (Ny)*, vol. 329, pp. 346–356, 2016, doi: 10.1016/j.ins.2015.09.037.
- [18] C. Coşkun, Y. Doç, and A. Baykal, “Comparison of classification algorithms in data mining on an example,” vol. 116, no. 22, pp. 51–58, 2011.