DAFTAR PUSTAKA

Alita, D. (2018). *ANALISIS PENGARUH EMOTICON DAN SARKASME PADA ANALISIS SENTIMEN BERBAHASA INDONESIA DI TWITTER*. Universitas Gajah Mada.

Blanchette, J. (2008). A Little Manual of API Design. *Oslo: Trolltech*.

Falahah, & Nur, D. (2015). *Pengembangan Aplikasi Sentiment Analysis Menggunakan Metode Naïve Bayes*.

Feldman, J., & Sanger, R. (2007). *The Text Mining Handbook: Advanced Apporoaches in Analysing Unstructured Data*. Cambridge University Press.

Gokgoz, E., & Subasi, A. (2015). Comparison Of Decision Tree Algorithms for EMG Signal Classification Using DWT. *Biomedical Signal Processing and Control*, *18*, 138–144.

Gusriani, S. (2016). *Analisis Sentimen Berdasarkan Komentar Publik Terhadap Toko Online Di Sosial Media Facebook (Studi Kasus: Zalora dan Berrybenka)*.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concept and Techniques Third Edition*. Elsevier Inc.

Hosmer, D. W., & Lemeshow, S. (2000). Applied Logistic Regression. *John Wiley and Sons*.

Hotho, A., Nurnberger, A., & Paass, G. (2005). A Brief Survey of Text Mining. *University of Kassel*.

Kemenkominfo. (2016). *Kementrian Informasi dan Informatika Republik Indonesia*. https://kominfo.go.id/index.php/content/detail/3415/kominfo+%3A+pengguna+internet+di+indonesia+63+juta+orang/0/berita_satker

Kotrika, R. (2016). Live Tweet MAP With Sentimental Analysis. *SUNY Polytechnic Institute Utica*.

Kurniawan, T. (2017). Implemntasi Text mining pada Analisis Sentimen Pengguna Twitter Menggunakan Naïve Bayes Classifier dan Support Vector Machine.

Institut Teknologi Sepuluh November.

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Morgan & Claypool publisher*.

Manning, C. D., Raghayan, P., & Schutze, H. (2009). *Introduction to Information Retrieval*. Cambridge University Press.

Mulsy, H. (2015). *Penerapan Metodedecision Tree Untuk Analisis Sentimen Pada Acara Televisi Indonesia*.

Nguyen, H., & Zheng, R. (2014). A Data-driven Study of Influences in Twitter International Chamber of Commerce. *IEEE*.

Nurhada, F. (2015). *Analisis Sentimen Masyarakat terhadap Calon Presiden Indonesia 2014 berdasarkan Opini dari Twitter Menggunakan Metode Naive Bayes Classifier*.

Olomouc. (2008). *Education technologi and senses inlearning*.

Patel, B., & Shah, D. (2013). Significance of stop word elimination in meta searh engine. *International Conference On Intelligent Systems and Signal Processing*, 52–55.

Pradesa, P. S. (2019). Analisis Sentimen pada Twitter tentang Kepuasaan Pelanggan Indihome dengan Metode Logistic Regression. *Universitas Telkom*.

Putri, D. R. A. (2018). *PENINGKATAN AKURASI PREDIKSI KLASIFIKASI PADA REGRESI LOGISTIK BINER MENGGUNAKAN ADAPTIVE BOOSTING*.

Rish, I. (2006). An Empirical Study of The Naive Bayes Classifier. *Classifier. International Joint Conference on Artifcial Intellegence*, 41–46.

Siang, J. (2005). Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB. *ANDI*.

Turban, E. (2011). Decision Support and Business Intelligence Systems. *Pearson Education*.

Yanti, W. N. (2018). Analisis Sentimen Media Sosial (Twitter) Terhadap Layanan Provider Telekomunikasi (Telkomsel) Menggunakan Metode Multinomial Naive Bayes. *Universitas Islam Negeri Maulana Malik Ibrahim Malang*.

LAMPIRAN

Lampiran 1. Data Hasil *TF-IDF*

	vaksin	gak	covid	kalau	jadi	…	sabtu	cegah	gk	gamau	ntar
0	0,000968	0	0,067814	0	0	…	0	0	0	0	0
1	0,002516	0	0	0	0	…	0	0	0	0	0
2	0,001797	0	0,251882	0	0	…	0	0	0	0	0
3	0,002795	0	0	0	0	…	0	0	0	0	0
4	0,001301	0,060004	0,060799	0,06451	0	…	0	0	0	0	0
5	0,001572	0	0,220397	0	0	…	0	0	0	0	0
6	0,000839	0	0,117545	0	0	…	0	0	0	0	0
7	0,002287	0	0	0	0	…	0	0	0	0	0
8	0,003431	0	0,160288	0	0	…	0	0	0	0	0
9	0,002516	0	0,352634	0	0	…	0	0	0	0	0
10	0,000968	0	0	0	0	…	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1990	0,002096	0	0	0	0	…	0	0	0	0	0
1991	0,000898	0	0,125941	0	0	…	0	0	0	0	0
1992	0,001398	0	0	0	0	…	0	0	0	0	0
1993	0,001572	0	0	0,23385	0,233445	…	0	0	0	0	0
1994	0,001797	0,248588	0	0	0	…	0	0	0	0	0
1995	0,002096	0	0,146931	0	0	…	0	0	0	0	0
1996	0,000699	0	0	0	0	…	0	0	0	0	0
1997	0,001144	0,105462	0	0,113382	0	…	0	0	0	0	0
1998	0,000812	0,168398	0	0	0,060244	…	0	0	0	0	0
1999	0,002096	0	0	0	0	…	0	0	0	0	0

Lampiran 2. *Syntax* Klasifikasi Data Menggunakan Python 3.7

```
# import Cross Validation Score Function

from sklearn.model_selection import cross_val_score


#import machine learning library

from sklearn.linear_model import LogisticRegression

from sklearn.naive_bayes import GaussianNB

from sklearn.naive_bayes import BernoulliNB

from sklearn.ensemble import RandomForestClassifier


import warnings

warnings.filterwarnings('ignore')

# model 1: Logistic Regression

print(cross_val_score(LogisticRegression(solver='newton-cg', multi_class='ovr'),
sentence_vectors, y, cv=3))

print(cross_val_score(LogisticRegression(solver='newton-cg', multi_class='ovr'),
tf_idf_model, y, cv=3))


scores = cross_val_score(LogisticRegression(solver='newton-cg',
multi_class='ovr'), tf_idf_model, y, cv=3)

print("Accuracy of Logistic Regression %0.2f" % (scores.mean()))


# model 2: Naive Bayes Classifier

print(cross_val_score(BernoulliNB(), sentence_vectors, y, cv=3))
```

```
print(cross_val_score(BernoulliNB(), tf_idf_model, y, cv=3))


scores = cross_val_score(BernoulliNB(), sentence_vectors, y, cv=3)
print("Accuracy of Naive Bayes Classifier %0.2f" % (scores.mean()))
```

Lampiran 3. Syntax Crawling Data Menggunakan Python

```
# import library
import tweepy
import pandas as pd

# variable untuk mengakses twitter API
ACCESS_TOKEN = 'your access secret'
ACCESS_SECRET = 'your access secret'
CONSUMER_KEY = 'your access secret'
CONSUMER_SECRET = 'your access secret'

# akses ke API
def connect_to_twitter_OAuth():
    auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
    auth.set_access_token(ACCESS_TOKEN,ACCESS_SECRET)
    api = tweepy.API(auth)
    return api

# membuat objek API
api = connect_to_twitter_OAuth()

# mengakses tweets berdasarkan hastag/keyword "data vaksin"
result = []
```

```
for tweet in tweepy.Cursor(api.search, q='vaksin|vaksinisasi -filter:retweets',
tweet_mode='Extended', lang='id', ).items(1000):

    result.append(tweet)

# mengubah list menjadi dataframe

def toDataFrame(tweets):

    DataSet = pd.DataFrame()

    DataSet['Tweet'] = [tweet.text for tweet in result]

    DataSet['userLocation'] = [tweet.user.location for tweet in tweets]

    return DataSet

DataSet = toDataFrame(result)

DataSet.to_excel('DataVaksin.xlsx')
```

Lampiran 4. *Syntax* Input dan Praproses Data Menggunakan Python 3.8

```
#import library

import pandas as pd

import re


from nltk.tokenize import word_tokenize

from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

import datetime, nltk, string

nltk.download("popular")

import warnings

warnings.filterwarnings('ignore')

data = pd.read_excel('DataVaksin covid.xlsx')

x = data['Tweet']

stemmer = StemmerFactory().create_stemmer()

stopwords = StopWordRemoverFactory().get_stop_words()                # default
stopwords

more_stopword = ['yg', 'daring', 'nya', 'kan', 'sih', 'pak', 'jg',

                 'dong', 'aja', 'kok', 'apa', 'dah', 'loh', 'nih',

                    'tuh', 'eh', 'lah', 'si']          # more stopword

stopwords = stopwords + more_stopword
```

```
def text_preprocess(tweet, stemmer, stopwords):

    for i in range(len(tweet)):

        tweet[i] = re.sub(r'@[\w:]+', '', tweet[i])         # menghilangkan @.....

        tweet[i] = re.sub(r'https?:\/\/\S+', '', tweet[i])  # menghilangkan tulisan https:..

        tweet[i] = remove_punctuation(tweet[i])             # menghilangkan karakter
tdk penting

        tweet[i] = tweet[i].lower()                         # mengubah semua kata menjadi
hurup kecil

        tweet[i] = re.sub('[0-9]+', '', tweet[i])

        tweet[i] = replace_words(tweet[i])

        tweet[i] = re.sub(r'\b[a-z]\b', '', tweet[i])

    tweet = tweet.apply(lambda x: ' '.join([stemmer.stem(item) for item in x.split()
if item not in stopwords]))

    return tweet

data['processed_text'] = text_preprocess(x, stemmer, stopwords)
```

Lampiran 5. Contoh Kasus Algoritma NBC

Contoh *tweet* pada gambar 2.3 sebagai data training dan contoh tweet "Ayo kita cegah dan tolak covid19" Sebagai data testing. Perhitungan dilakukan untuk mengklasifikasikan apakah contoh positif atau negatif. Hal pertama yang dilakukan adalah menghitung probabilitas setiap kelas sentimen dengan persamaan (2.4). sebagai berikut.

$$P(v_1) = \frac{|doc\ 1|}{|training|} = \frac{1}{2} = 0{,}5$$

$$P(v_2) = \frac{|doc\ 2|}{|training|} = \frac{1}{2} = 0{,}5$$

Dimana $P(v_1)$ adalah probabilitas sentimen positif dan $P(v_2)$ adalah probabilitas sentimen negatif. Kemudian dilakukan perhitungan probabilitas kemunculan setiap kata pada masing-masing kategori dengan persamaan (2.5) Seperti berikut:

$P(ayo \mid v_1)$ = (1+1)/(9+10) = 0,1052

$P(ayo \mid v_2)$ = (1+1)/(4+10) = 0,1428

$P(ingat \mid v_1)$ = (1+1)/(9+10) = 0,1052

$P(ingat \mid v_2)$ = (0+1)/(4+10) = 0,0714

$P(untuk \mid v_1)$ = (1+1)/(9+10) = 0,1052

$P(untuk \mid v_2)$ = (0+1)/(4+10) = 0,0714

$P(syukur \mid v_1)$ = (1+1)/(9+10) = 0,1052

$P(syukur \mid v_2)$ = (0+1)/(4+10) = 0,0714

$P(vaksin \mid v_1)$ = (1+1)/(9+10) = 0,1052

$P(vaksin \mid v_2)$ = (1+1)/(4+10) = 0,1428

$P(siap \mid v_1)$ = (1+1)/(9+10) = 0,1052

$P(siap|\ v_2)$ = (0+1)/(4+10) = 0,0714

$P(cegah|\ v_1)$ = 1+1)/(9+10) = 0,1052

$P(cegah|\ v_2)$ = (0+1)/(4+10) = 0,0714

$P(nular\ |\ v_1)$ = (1+1)/(9+10) = 0,1052

$P(nular\ |\ v_2)$ = (0+1)/(4+10) = 0,0714

$P(covid19\ |\ v_1)$ = (1+1)/(9+10) = 0,1052

$P(covid19\ |\ v_2)$ = (1+1)/(4+10) = 0,1428

$P(tolak\ |\ v_1)$ = (0+1)/(9+10) = 0,0526

$P(tolak\ \ |\ v_2)$ = (1+1)/(4+10) = 0,1428

Selanjutnya adalah mencari probabilitas tertinggi dari *tweet* yang diujikan. *Tweet* testing setelah dilakukan praproses teks., maka terdiri dari kata "ayo", "cegah", "tolak" dan "covid19". Sehingga dicari probabilitas tertinggi dari setiap kata pada tweet tersebut menggunakan persamaan (2.3).

$\mathrm{P}(v_1)\ \prod_i P(a_i|v_1)$ = $(0{,}5)(P(ayo|v_1)\ \times\ P(cegah|v_1) \times\ P(tolak|v_1) \times P(covid19|v_1)$

$=(0{,}5)\ (0{,}1052 \times 0{,}1052 \times 0{,}0526 \times 0{,}1052)$

$= 0{,}5\ \times 0{,}0000612$

$= 0{,}0000306$

$\mathrm{P}(v_2)\ \prod_i P(a_i|v_2)$ == $(0{,}5)(P(ayo|v_2)\ \times\ P(cegah|v_2) \times P(tolak|v_2) \times P(covid19|v_2)$

$=(0{,}5)\ (0{,}1428 \times 0{,}0714 \times 0{,}1428 \times 0{,}1428\ )$

$= 0{,}5\ \times 0{,}000207$

$= 0{,}000103$

$$V_{MAP} = \arg\max_{v_j=V} \mathrm{P}(v_j) \prod_i P(a_i|v_j) = v_2$$

Nilai probabilitas kata setiap tweet testing yang terbesar adalah probabilitas setiap kata pada sentimen negatif sehingga tweet testing tersebut diklasifikasikan sebagai tweet dengan sentimen negatif.

Lampiran 6. *Syntax Word Cloud* Menggunakan Python 3.8

```
# visualization with word cloud data Pro
words = ''.join([word for word in data['processed_text'][data['Label']=='Pro']])
wordCloud = WordCloud(background_color = 'white').generate(words)

fig = plt.figure()
fig.set_figwidth(8)
fig.set_figheight(8)
plt.imshow(wordCloud, interpolation='bilinear')
plt.axis('off')
plt.show()
fig.savefig('Sentiment Pro.jpg')

# visualization with word cloud data Kontra
words = ''.join([word for word in data['processed_text'][data['Label']=='Kontra']])
wordCloud = WordCloud(background_color = 'white').generate(words)
fig = plt.figure()
```

```
fig.set_figwidth(8)

fig.set_figheight(8)

plt.imshow(wordCloud, interpolation='bilinear')

plt.axis('off')

plt.show()

fig.savefig('Sentiment Kontra.jpg')
```

Lampiran 7. Tabel *Chi-Kuadrat*

DF	Probabilitas				
	0,5	0,1	0,05	0,01	0,05
1	0,45494	2,70554	3,84146	6,63490	3,84146
2	1,38629	4,60517	5,99146	9,21034	5,99146
3	2,36597	6,25139	7,81473	11,34487	7,81473
4	3,35669	7,77944	9,48773	13,27670	9,48773
5	4,35146	9,23636	11,07050	15,08627	11,07050
6	5,34812	10,64464	12,59159	16,81189	12,59159
7	6,34581	12,01704	14,06714	18,47531	14,06714
8	7,34412	13,36157	15,50731	20,09024	15,50731
9	8,34283	14,68366	16,91898	21,66599	16,91898
10	9,34182	15,98718	18,30704	23,20925	18,30704
⋮	⋮	⋮	⋮	⋮	⋮
190	189,3338	215,3711	223,1602	238,2664	223,1602
191	190,3337	216,4368	224,2446	239,3856	224,2446
192	191,3337	217,5024	225,3288	240,5046	225,3288
193	192,3337	218,5678	226,4127	241,6232	226,4127
194	193,3337	219,6331	227,4964	242,7415	227,4964
195	194,3337	220,6981	228,5799	243,8595	228,5799
196	195,3337	221,7631	229,6632	244,9772	229,6632
197	196,3337	222,8278	230,7463	246,0947	230,7463
198	197,3337	223,8924	231,8292	247,2118	231,8292
199	198,3337	224,9568	232,9118	248,3286	232,9118
200	199,3337	226,021	233,9943	249,4451	233,9943

Lampiran 8. Output Program

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	655.908	200	.000
	Block	655.908	200	.000
	Model	655.908	200	.000

Hosmer and Lemeshow Test

Step	Chi-square	Df	Sig.
1	7.372	8	.497

Classification Table[a]

			Predicted		
			target		Percentage
	Observed		0	1	Correct
Step 1	target	0	503	343	59.5
		1	205	949	82.2
	Overall Percentage				72.6

a. The cut value is .500