

ANALISIS PERBANDINGAN SENTIMEN CORONA VIRUS DISEASE-2019 (COVID19) PADA TWITTER MENGGUNAKAN METODE LOGISTIC REGRESSION DAN SUPPORT VECTOR MACHINE (SVM)

Kelvin, Jepri Banjarnahor, Evta Indra, Stiven Hamonangan Sinurat.
Sistem Informasi, FTIK, Universitas Prima Indonesia
Jalan Sampul, Medan
E-mail : jepribanjarnahor@unprimdn.ac.id

ABSTRAK- Melihat perkembangan twitter tersebut maka twitter menjadi salah satu media yang dapat digunakan untuk melakukan analisis sentimen terhadap berbagai topik. Penelitian ini melakukan suatu analisis sentimen terhadap bahasan yang saat ini sering menjadi trending topic di twitter yaitu “CoronaVirus Disease-2019 (covid19)”. Penyebaran virus ini juga langsung dibicarakan oleh banyak kalangan masyarakat twitter, saat ini virus corona tengah menjadi perhatian dunia internasional. Banyaknya jumlah angka korban dan cepatnya penularan virus membuat masyarakat khawatir dan muncul berbagai opini tentang virus corona, Opini inilah yang kemudian di analisa untuk diketahui polaritasnya dengan analisis sentimen. Metode yang digunakan adalah Logistic Regression dan Support Vector Machine (SVM) dimana SVM memiliki nilai akurasi 91,15% dalam data test sedangkan metode Logistic Regression mendapatkan nilai akurasi sebanyak 87,68% dalam data test.

Kata kunci : Sentimen Analisis; *NLP*; *Logistic Regression*; *Support Vector Machine (SVM)*;

1. PENDAHULUAN

Perkembangan media online yang begitu pesat, memunculkan banyak media mulai dari media berita sampai media sosial. Media sosial saja sudah begitu banyak, dari Facebook, Twitter, Path, Instagram, Google+, Tumblr, LinkedIn dan masih banyak lagi[1]. Pengguna aktif harian twitter di indonesia saat ini cukup banyak, saat ini diberbagai belahan dunia twitter diklaim menjadi salah satu media online yang pengguna aktif harian yang paling banyak berdasarkan laporan finansial twitter pada kuartal ke-3 tahun 2019[2]. Melihat perkembangan twitter tersebut maka twitter menjadi salah satu media yang dapat digunakan untuk melakukan analisis sentimen terhadap berbagai topik. Penelitian ini melakukan suatu analisis sentimen terhadap bahasan yang saat ini sering menjadi trending topic di twitter yaitu “CoronaVirus Disease-2019 (covid19)”.

COVID-19 adalah penyakit menular yang disebabkan oleh jenis coronavirus yang baru ditemukan. Ini merupakan virus baru dan penyakit yang tidak dikenal sebelum terjadi wabah di Wuhan, Tiongkok, bulan Desember 2019[3]. Penyebaran virus ini juga langsung dibicarakan oleh banyak kalangan masyarakat twitter, saat ini virus corona tengah menjadi perhatian dunia internasional. Banyaknya jumlah angka korban dan cepatnya penularan virus membuat masyarakat khawatir dan muncul berbagai opini tentang virus corona, Opini inilah yang kemudian di analisa untuk diketahui polaritasnya dengan analisis sentimen.

Penelitian menggunakan machine learning untuk mengklasifikasikan movie reviews (Pang, 2002). Penelitian ini pernah dilakukan dengan cara mengklasifikasikan sentimen terhadap review film dan menentukan apakah review film tersebut

memiliki sentimen positif ataukah negatif. Fitur yang berbeda dari review tersebut diekstrak dan digunakan algoritma machine learning Naïve Bayes dan Support Vector Machine (SVM) untuk menghasilkan model klasifikasi. Mereka memperoleh hasil akurasi antara 78,7% ketika menggunakan Naïve Bayes dan nilai akurasi yang diperoleh ketika menggunakan SVM adalah 72,8%.

Berdasarkan latar belakang yang telah dipaparkan, penulis tertarik untuk melakukan penelitian dengan judul “**Analisis perbandingan sentimen Corona Virus Disease-2019 (Covid19) pada Twitter Menggunakan Metode Logistic Regression Dan Support Vector Machine (SVM)**”.

2. LANDASAN TEORI

2.1 Jenis Penelitian

Sentiment analisis digunakan untuk mengetahui sentiment atau polarity sebuah teks apakah Extremely positive , positive, neutral, negative, Extremely negative. Biasanya sentiment analysis diterapkan pada data teks opini masyarakat terhadap suatu objek, misalkan review dari suatu produk e-commerce, review sebuah film, dan komentar yang terdapat pada sosial media.

Media sosial khususnya Twitter sekarang ini menjadi perangkat komunikasi yang sangat populer di kalangan pengguna internet. Pada konferensi resmi pengembang Twitter Chirp 2010, perusahaan tersebut menyampaikan statistik mengenai situs dan pengguna Twitter. Statistik tersebut menyebutkan bahwa pada bulan April 2010, Twitter memiliki 106 juta akun dan sebanyak 180 juta pengunjung unik setiap bulannya. Jumlah pengguna Twitter disebutkan terus meningkat 300.000 user setiap harinya.[4]

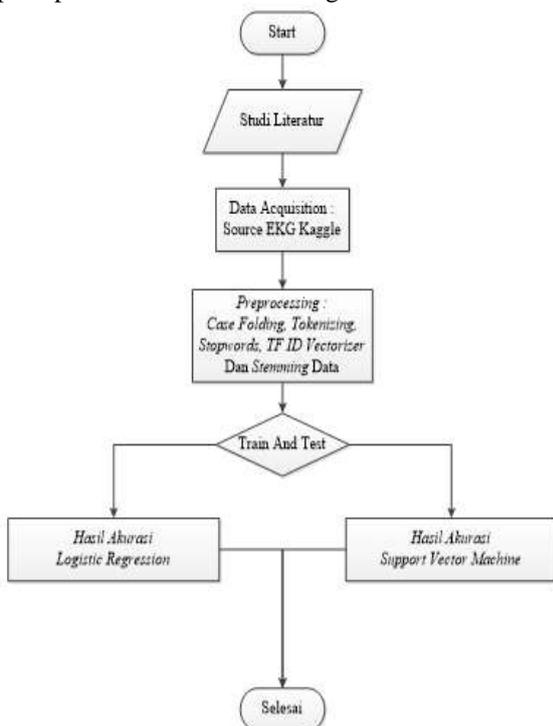
Pengguna Twitter bebas untuk mengeluarkan pendapat dan opininya, termasuk opini mengenai pandemi Corona Virus. Penelitian ini menampung opini masyarakat dengan membagi menjadi lima class opini yaitu extremely positif, positif, netral, negatif dan extremely negatif. Proses klasifikasi pada penelitian ini menggunakan metode klasifikasi Logistic Regression dan Support Vector Machine dengan preprocessing data menggunakan case folding, tokenizing, stopwords, stemming, dan TF IDF Vectorizer. Data yang digunakan adalah tweet dalam bahasa Inggris tentang calon Corona Virus, dengan jumlah dataset sebanyak 3798 tweet yang didistribusikan secara merata kedalam lima class opini.

Logistic regression adalah model statistik yang digunakan untuk menentukan apakah sebuah independent variable memiliki pengaruh terhadap sebuah binary dependent variable. Dengan menggunakan sigmoid function, logistic regression menghasilkan output sebuah probability antara angka 0 dan 1.[5]

Support Vector Machine (SVM) adalah seperangkat metode pembelajaran terbimbing yang menganalisis data dan mengenali pola, digunakan untuk klasifikasi dan analisis regresi. Algoritma SVM asli diciptakan oleh Vladimir Vapnik dan turunan standar saat ini (margin lunak) diusulkan oleh Corinna Cortes dan Vapnik Vladimi.[6]

2.2 Prosedur Kerja

Agar penelitian dapat berjalan dengan baik selesai dengan tepat waktu maka terdapat prosedur kerja penelitian. Adapun prosedur kerja penelitian pada penelitian ini adalah sebagai berikut :



Gambar 1. Flowchart

3. HASIL DAN PEMBAHASAN

3.1 Analisis Masalah

Sentiment analysis digunakan untuk mengetahui sentiment atau polarity sebuah teks apakah positive, neutral, atau negative. Biasanya sentiment analysis diterapkan pada data teks opini masyarakat terhadap suatu objek, misalkan review dari suatu produk e-commerce, review sebuah film, dan komentar yang terdapat pada sosial media.

3.2 Analisis Data

Sentiment analysis termasuk yang banyak dipelajari ketika belajar natural language processing, karena ketersediaan data yang melimpah dan mudah didapatkan berdasarkan dari sumbernya (misal: sosial media). Untuk melakukannya natural language processing membutuhkan dataset yang digunakan untuk melatih. Dataset yang digunakan dalam studi ini bersumber dari : [7]

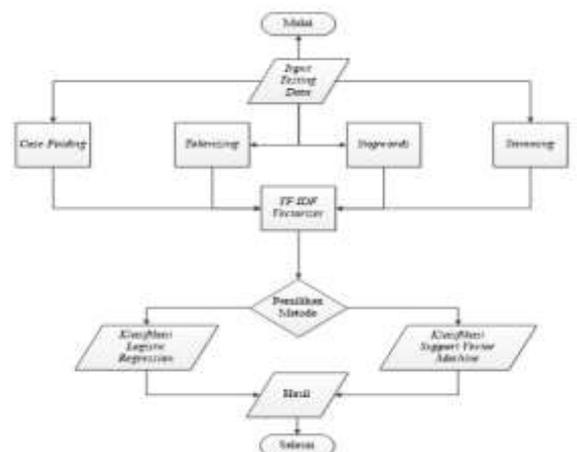
Dataset tersebut berupa data tweet yang diambil dari 3798 subjek yang terdiri dari 592 Extremely positive, 947 positive, 619 neutral, 592 negative, 1041 Extremely negative.

	Original Tweet	Sentiment
0	trending new yorkers encounter empty supermarket	Extremely Negative
1	when i couldn't find hand sanitizer at fred me	Positive
2	find out how you can protect yourself and love	Extremely Positive
3	panic buying hits newyork city as anxious shop	Negative
4	toiletpaper dunnypaper coronavirus coronavirus	Neutral
...		
95	the government must provide hand sanitizer in	Extremely Positive
96	what you need if quarantined at home coronavir	Neutral
97	see the new fuji film x-14 and x100v at robe	Extremely Positive
98	spiking prices during a state of emergency is	Extremely Negative
99	besides canned food and toilet paper hitps://coo	Neutral

Gambar 2. Dataset Twitter

3.3 Pengolahan Data

Tahapan proses pengolahan data yang dilakukan dalam Machine Learning ini digambarkan dalam sebuah alur metodologi penelitian seperti tertuang pada gambar 3.2.



Gambar 3. Flowchart Machine Learning

3.3.1 Casefolding

Case folding adalah salah satu bentuk *text preprocessing* yang paling sederhana dan efektif meskipun sering diabaikan. Tujuan dari *case folding* untuk mengubah semua huruf dalam dokumen menjadi huruf kecil, menghapus nomor yang tidak relevan di dataset, menghapus tanda baca seperti `[!'"#$%&'()*+,-./:;<=>@[\\]^_`{|}~]`. Pada tahap ini tidak menggunakan *external library* apapun, kita bisa memanfaatkan modul yang tersedia di python.

```
Source Code :
#CaseFolding
kalimat = "What Precautionary measures have you all taken in your respective Restaurants and Hotels, Comment below... #COVID-19 #Coronavirus COVID-19: Hotel chains enforce precautionary measures, issue safety, hygiene advisories"
kalimat = kalimat.lower()
kalimat = kalimat.strip(" ")
kalimat = re.sub(r'[?|!|,|_|#|@|/|#|"](-+)', '', kalimat)
casefolding = kalimat.split()
print(casefolding)
```

Tabel 1. Proses Case Folding

Tweet awal	Proses	Tweet Hasil
What Precautionary measures have you all taken in your respective Restaurants and Hotels, Comment below... #COVID-19 #Coronavirus COVID-19: Hotel chains enforce precautionary measures, issue safety, hygiene advisories	Case Folding	what precautionary measures have you all taken in your respective restaurants and hotels comment below covid-19 coronavirus covid-19 hotel chains enforce precautionary measures issue safety hygiene advisories

3.3.2 Tokenizing

Proses untuk memisahkan input text menjadi potongan yang memiliki arti disebut tokenization. Hasil dari proses tokenization disebut token. Token dapat berupa kata, kalimat, paragraph dan lainnya.

Source Code:

```
#Tokenizing
kalimat = "what precautionary measures have you all taken in your respective restaurants and hotels comment below covid-19 coronavirus covid-19 hotel chains enforce precautionary measures issue safety hygiene advisories"

pisah = kalimat.split()
print(pisah)
```

Tabel 2 Proses Tokenizing

Tweet awal	Proses	Tweet Hasil
what precautionary measures have you all taken in your respective restaurants and hotels comment below covid-19 coronavirus covid-19 hotel chains enforce precautionary measures issue safety hygiene advisories	Tokenizing	['what', 'precautionary', 'measures', 'have', 'you', 'all', 'taken', 'in', 'your', 'respective', 'restaurants', 'and', 'hotels', 'comment', 'below', 'covid-19', 'coronavirus', 'covid-19', 'hotel', 'chains', 'enforce', 'precautionary', 'measures', 'issue', 'safety', 'hygiene', 'advisories']

3.3.3 Stopwords

Stopwords adalah tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting)

atau *wordlist* (menyimpan kata penting). *Stopword* adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna.

Source Code:

```
#stopwords
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
kalimat = "what precautionary measures have you all taken in your respective restaurants and hotels comment below covid-19 coronavirus covid-19 hotel chains enforce precautionary measures issue safety hygiene advisories."

tokens = word_tokenize(kalimat)
listStopword = set(stopwords.words('english'))

removed = []
for t in tokens:
    if t not in listStopword:
        removed.append(t)

print(removed)
```

Tabel 3. Proses Stopwords

Tweet awal	Proses	Tweet Hasil
what precautionary measures have you all taken in your respective restaurants and hotels comment below covid-19 coronavirus covid-19 hotel chains enforce precautionary measures issue safety hygiene advisories	Stopwords	'precautionary', 'measures', 'taken', 'respective', 'restaurants', 'hotels', 'comment', 'covid-19', 'coronavirus', 'covid-19', 'hotel', 'chains', 'enforce', 'precautionary', 'measures', 'issue', 'safety', 'hygiene', 'advisories', ''

3.3.4 Stemming

Stemming yaitu melakukan proses mencari kata dasar dari setiap kata hasil proses filtering sebelumnya. Stemming melakukan proses pengembalian berbagai bentuk kata menjadi bentuk kata dasar dengan menghilangkan imbuhan. Berikut merupakan hasil data komentar yang telah di Stemming.

Source Code :

```
#stemming
stemmer = nltk.stem.PorterStemmer()
" ".join(stemmer.stem(token) for token in tokens)
```

Tabel 4. Proses Stemming

Tweet awal	Proses	Tweet Hasil
what precautionary measures have you all taken in your respective restaurants and hotels comment below covid-19 coronavirus covid-19 hotel chains enforce precautionary measures issue safety hygiene advisories	Stemming	what precautionari measur have you all taken in your respect restaur and hotel comment below covid-19 coronavirus covid-19 hotel chain enforce precautionari measur issu safeti higien advisori

3.3.5 TF IDF Vectorizer

Proses pembobotan kata adalah proses mengubah kata menjadi bentuk angka atau vector, sedangkan TF (term frekuensi) merupakan untuk menentukan nilai frekuensi sebuah kata di dalam sebuah dokumen dan pemberian bobot pada setiap kata pada

setiap dokumen untuk mencari dan menghitung berapa kali kata itu muncul.

Source Code:

```
a = "what precautionari measur have you all
taken in your respect restaur and hotel comm
ent below covid-19 coronaviru covid-
19 hotel chain enforc precautionari measur i
ssu safeti hygien advisori".split()
docs = [a,b,c]
def tfidf(word, sentence):
    # term frequency
    tf = sentence.count(word) / len(sentence)
)
    # inverse document frequency
    idf = np.log10(len(docs) / sum([1 for do
c in docs if word in doc]))
    return round(tf*idf, 4)
tfidf('what','precautionary','have','you','a
ll','taken','in','your','respect','restauran'
, a)
```

Tabel 5. Proses TF-IDF Vectorizer

No	Term	TF-IDF
1	what	0.0177
2	precautionary	0.0353
3	have	0.0177
4	you	0.0065
5	all	0.0177
6	taken	0.0177
7	in	0.0177
8	your	0.0065
9	respect	0.0177
10	restauran	0.0177

3.4 Pengolahan Machine Learning

Machine learning merupakan serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara mempresentasikan data-data tersebut dengan algoritma pembelajaran[8]. pengolahan kedalam machine learning Logistic regression Dan Support Vector Machine sebelum melakukan training dan test. Dalam penelitian pembagian data training dan test adalah 80% : 20%.

a. Accuracy

Merupakan rasio prediksi Benar (extremely positif, positif, netral, negatif dan extremely negatif) dengan keseluruhan data. Akurasi = (TP + TN) / (TP+FP+FN+TN)

b. Precision

Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Precision = (TP) / (TP+FP)

c. Recall

Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Recall = (TP) / (TP + FN)

d. F1 Score

F1 Score merupakan perbandingan rata-rata presisi dan recall yang dibobotkan. F1 Score = 2 * (Recall*Precision) / (Recall + Precision).

3.4.1 Logistci Regression

Menurut [9] tujuan melakukan analisis data kategori menggunakan regresi logistik adalah mendapatkan model terbaik dan sederhana untuk menjelaskan hubungan antara keluaran dari variabel respons (Y) dengan variabel-variabel prediktornya (X). Variabel respons dalam regresi logistik dapat berupa kategori atau kualitatif, sedangkan variabel prediktornya dapat berupa kualitatif dan kuantitatif. Jika variabel Y merupakan variabel biner atau dikotomi dalam artian variabel respons terdiri dari dua kategori yaitu “sukses” (Y = 1) atau “gagal” (Y = 0), maka variabel Y mengikuti sebaran Bernoulli yang memiliki fungsi densitas peluang.

Rumus Logistic Regression :

$$Accuracy = \frac{\sum_{i=1}^m (y_{pred}^{(i)} == y_{true}^{(i)})}{m}$$

Dimana :

m = jumlah data test.

Source Code :

```
log_reg = LogisticRegression().fit(X, Y)
#predict on train
train_predsLog = log_reg.predict(X_train)
#accuracy on train

#predict on test
test_predsLog = log_reg.predict(X_test)
#accuracy on test
print("confusion_matrix train is: ", confusi
on_matrix(Y_train, train_predsLog))
print('-'*50)
print("confusion_matrix test is: ", confusio
n_matrix(Y_test, test_predsLog))
print('-'*50)
print("Model accuracy on train is: ", accura
cy_score(Y_train, train_predsLog))
print("Model accuracy on test is: ", accurac
y_score(Y_test, test_predsLog))
print(classification_report(Y_test, test_pred
sLog))
```

```
confusion_matrix train is: [[340  0  64  8 14]
 [ 4 365  31  4 45]
 [ 4  5 753  6 19]
 [ 5  3  49 376  32]
 [ 4  6  34  7 670]]
-----
confusion_matrix test is: [[128  1  28  0  9]
 [ 1 122  10  2 15]
 [ 3  1 246  0  4]
 [ 2  0  17 124 11]
 [ 1  2  9  1 213]]
-----
Model accuracy on train is: 0.8792134831460674
Model accuracy on test is: 0.8768421052631579
```

Gambar 4. Hasil Akurasi Logistic Regression

Pada konversi matrix dengan metode Logistic Regression, dapat diperoleh hasil akurasi analisis sentimen Twitter sebanyak 87,79% dalam data training dan 87,68% dalam data test. Untuk hasil

akurasi precision, recall, f1-score 5 parameter sentimen dapat dilihat pada tabel 3.6:

Tabel 6. Detail Akurasi *Logistic Regression*

	precision	recall	f1-score	support
Extremely Negative	0.95	0.77	0.85	166
Extremely Positive	0.97	0.81	0.88	150
Negative	0.79	0.97	0.87	254
Neutral	0.98	0.81	0.88	154
Positive	0.85	0.94	0.89	226
accuracy			0.88	950
macro avg	0.91	0.86	0.88	950
weighted avg	0.89	0.88	0.88	950

3.4.2 Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu metode dalam *supervised learning* yang biasanya digunakan untuk klasifikasi (seperti *Support Vector Classification*) dan regresi (*Support Vector Regression*). Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan *linear* maupun *non linear*.

$$f(x_d) = \sum_{i=1}^{Nsv} a_{iyi} (yx_i^T x + r)^p + b$$

Keterangan :

- Nsv = jumlah *Support Vector*
- a = *alpha*, pengali *Lanrange*
- i = 1,2,3, ..., Nsv
- y = label / kelas dari data
- b = bias

$(yx_i^T x + r)^p + b$ = persamaan polynominal linear

Source Code:

```
#fit the model on train data
SVM = SVC(kernel='linear')
SVM.fit(X, Y)

#predict on train
train_preds5 = SVM.predict(X_train)
#accuracy on train

#predict on test
test_preds5 = SVM.predict(X_test)
#accuracy on test
print('-'*50)

#Confusion matrix
print("confusion_matrix train is: ", confusi
on_matrix(Y_train, train_preds5))

print("confusion_matrix test is: ", confusio
n_matrix(Y_test, test_preds5))
print('-'*50)
print("Model accuracy on train is: ", accura
cy_score(Y_train, train_preds5))
print("Model accuracy on test is: ", accurac
y_score(Y_test, test_preds5))
print('-'*50)
```

```
print(classification_report(Y_test, test_pred
s5))
```

```
confusion_matrix train is: [[377  1  37  3  8]
 [ 4 400 13  1 31]
 [ 5  2 762  2 16]
 [ 4  1 31 407 22]
 [ 4  5 30  4 678]]
confusion_matrix test is: [[142  0 19  1  4]
 [ 0 131  6  1 12]
 [ 3  1 245  0  5]
 [ 2  0 14 131  7]
 [ 2  2  5  0 217]]
```

Model accuracy on train is: 0.9213483146067416

Model accuracy on test is: 0.911578947368421

Gambar 5. Hasil Akurasi SVM

Pada konversi matrix dengan metode *Support Vector Machine*(SVM), dapat diperoleh hasil akurasi analisis sentimen *Twitter* sebanyak 92,13% dalam data *training* dan 91,15% dalam data *test*. Untuk hasil akurasi precision, recall, f1-score 5 parameter sentimen dapat dilihat pada tabel 3.7:

Tabel 7. Detail Akurasi SVM

	precision	recall	f1-score	support
Extremely Negative	0.95	0.86	0.90	166
Extremely Positive	0.98	0.87	0.92	150
Negative	0.85	0.96	0.90	254
Neutral	0.98	0.85	0.91	154
Positive	0.89	0.96	0.92	226
accuracy			0.91	950
macro avg	0.93	0.90	0.91	950
weighted avg	0.92	0.91	0.91	950

3.4.3 Perbandingan Akurasi

Setelah dilakukan semua tahap untuk menganalisis sentimen *tweet* tentang kasus *Covid-19* dengan menggunakan metode *Logistic Regression* dan *Support Vector Machine*(SVM) diperoleh akurasi 87,68% untuk *Logistic Regression* dan 91,15% untuk *Support Vector Machine*(SVM) maka diperoleh rentan perbedaan nilai akurasi sebesar 3,47%. Dapat kita lihat pada Gambar 3.5.

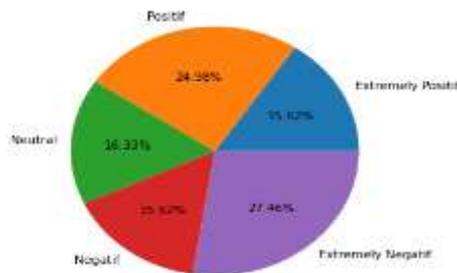
<pre> confusion matrix train size: [[340 0 64 0 14] [4 365 0 0 0] [4 5 763 0 13] [5 3 49 374 11] [4 4 34 7 470]] </pre>	<pre> confusion matrix train size: [[377 1 37 0 0] [4 401 13 1 11] [5 2 762 2 13] [4 1 31 407 11] [4 5 23 4 470]] </pre>
<pre> confusion matrix test size: [[129 1 28 0 0] [1 122 11 2 13] [3 1 244 0 0] [3 0 17 124 11] [1 2 5 1 211]] </pre>	<pre> confusion matrix test size: [[142 0 19 0 0] [1 131 6 1 11] [3 1 245 0 0] [2 0 14 131 11] [2 2 5 0 211]] </pre>
<pre> Model accuracy on train size: 0.976214601460146 Model accuracy on test size: 0.97494102063579 </pre>	<pre> Model accuracy on train size: 0.921346114607146 Model accuracy on test size: 0.91157947468421 </pre>

Gambar 6. Hasil Perbandingan Akurasi

3.5 Visualisasi Data

3.5.1 Pie Chart

Pie chart atau diagram lingkaran merupakan grafik statistik berbentuk lingkaran yang dibagi menjadi beberapa irisan dan luasnya bergantung kepada proporsi numerik atau kuantitas dari data yang dimiliki[10]. Satu lingkaran menunjukkan bagian utuh atau seratus persen. Setengah lingkaran menunjukkan proporsi setengah dari total atau lima puluh persennya, dan seterusnya. Dengan total data yang diambil dari 3798 subjek yang terdiri dari 592 *Extremely positive*, 947 *positive*, 619 *neutral*, 592 *negative*, 1041 *Extremely negative* dapat dilihat pada gambar 3.6.



Gambar 7. Pie Chart Hasil

3.6 Hasil Akhir

Hasil akhir dari penelitian yang dilakukan adalah melihat perbandingan akurasi dari kedua metode yang digunakan dalam menganalisis sentimen *Twitter*. Dari kedua metode yang digunakan *Support Vector Machine* (SVM) menghasilkan nilai akurasi yang paling tinggi sebanyak 92,13% dalam data *training* dan 91,15% dalam data *test* sedangkan metode *Logistic Regression* mendapatkan nilai akurasi sebanyak 87,79% dalam data *training* dan 87,68% dalam data *test*. Dari nilai akurasi yang didapatkan maka hasil akhir dari penelitian ini adalah metode *Support Vector Machine*(SVM) unggul dalam melakukan klasifikasi terhadap sentimen *Twitter* tentang kasus *Covid-19*.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Banyaknya jumlah angka korban dan cepatnya penularan virus membuat masyarakat khawatir dan muncul berbagai opini tentang virus corona, Opini inilah yang kemudian di analisa untuk diketahui polaritasnya dengan analisis sentimen.

Dari analisis yang telah dilakukan dengan tahapan yang ada dapat disimpulkan dari penelitian ini adalah nilai akurasi metode *Support Vector Machine*(SVM) lebih unggul dibandingkan metode *Logistic Regression* dalam melakukan klasifikasi terhadap sentimen *Twitter* tentang *Tweet* kasus *CoronaVirus Disease-2019* (*Covid-19*) dengan nilai akurasi sebesar 91,15%.

4.2 Saran

Pada penelitian ini analisis yang dilakukan hanya sebatas *Tweet* dengan 5 sentiment atau polarity sebuah teks apakah *Extremely positive* , *positive*, *neutral*, *negative*, *Extremely negative*.Diharapkan pada peneliti selanjutnya diharapkan dapat membuat rating bintang terhadap komentar yang dianalisis.

DAFTAR PUSTAKA

- [1] Top Media Sosial <http://www.evadollzz.com/2014/09/top-10-social-networkings-terpopuler.html>
- [2] Annual Report Twitter, 2019. <https://investor.twitterinc.com/financial-information/annual-reports/default.aspx>
- [3] <https://covid19.go.id/>
- [4] M. Yarrow, J. Clausen, and P. Robbins, "The social meaning of mental illness," *J. Soc. Issues*, pp. 443–454, 2010.
- [5] <https://medium.com/@yasirabd/sentiment-analysis-dengan-logistic-regression-50315cd2c836>
- [6] Putranti, N.D., dan Winarko, E. (2014). "Analisis Sentimen Twitter untuk Teks Berbahasa. Indonesia dengan Maximum Entropy dan Support Vector Machine
- [7] Danial dan Wasriah. (2009:80). Metode Penulisan Karya Ilmiah. Bandung: Laboratorium Pendidikan Kewarganegaraan UPI.
- [8] <https://www.kaggle.com/datatattle/covid-19-nlp-text-classification>
- [9] Danukusumo, K. P. (2017). Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU.
- [10] <https://medium.com/the-mission/to-pie-charts-3b1f57bcb34a>