

Analisis Sentimen Pengguna Twitter terhadap Perpanjangan PPKM Menggunakan Metode *K-Nearest Neighbor*

Arief Asro'i^[1]; Herny Februariyanti^[2]

Sistem Informasi Universitas Stikubank Semarang ^{[1][2]}

*Corresponding author: hernyfeb@edu.unisbank.ac.id²

INFO ARTIKEL	INTISARI
<p>Diajukan : 20 Maret 2022</p> <p>Diterima : 30 Maret 2022</p> <p>Diterbitkan: 25 Juni 2022</p> <p>Kata Kunci : Sentimen, Twitter, PPKM, <i>K-Nearest Neighbor</i></p> <p>Keyword : Sentiment, Twitter, PPKM, <i>K-Nearest Neighbor</i></p>	<p>Banyaknya berita buruk yang beredar dalam sosial media mengenai kebijakan PPKM yang terus-menerus diperpanjang menggugah rasa penasaran peneliti untuk memastikan apakah benar tanggapan masyarakat mengenai PPKM yang terus-menerus diperpanjang memanglah menuai kesan negatif. Dengan demikian, peneliti melakukan analisis sentimen pengguna twitter terhadap perpanjangan PPKM yang dapat digunakan sebagai bahan evaluasi dalam menentukan kebijakan. Twitter digunakan sebagai sumber data karena baru-baru ini twitter sedang naik daun setelah bertahun-tahun pasif karena kalah bersaing dengan sosial media lain. Dengan memanfaatkan teknologi machine learning kita dapat mengetahui sentimen seseorang berdasarkan ilmu statistik yang telah dikombinasikan dengan programming. Peneliti menggunakan algoritma <i>K-Nearest Neighbor</i> untuk menentukan sentimen pengguna twitter dengan bantuan <i>library Scikit-learn</i> yang populer di kalangan Data Scientist. Algoritma tersebut diterapkan ke 6408 data tweet dengan kata kunci "PPKM" yang dikumpulkan pada 1 Juli 2021 – 31 Desember 2021. Hasil training model membuktikan bahwa skor akurasi 69,5%, recall 69,5% , dan presisi 68,7%.</p> <p><i>Abstract - The amount of bad news circulating on social media regarding the PPKM policy being continuously extended has aroused the curiosity of researchers to ascertain whether it is true that the public's response to the PPKM being continuously extended is indeed reaping a negative impression. Thus, the researcher conducted an analysis of twitter users' sentiment towards the extension of the PPKM which could be used as an evaluation material in determining policies. Twitter is used as a data source because recently twitter is on the rise after years of being passive due to being unable to compete with other social media. By utilizing machine learning technology, we can find out someone's sentiments based on statistical knowledge that has been combined with programming. The researcher uses the K-Nearest Neighbor algorithm to determine the sentiment of twitter users with the help of the Scikit-learn library which is popular among Data Scientists. The algorithm is applied to 6408 tweet data with the keyword "PPKM" collected on July 1, 2021 - December 31, 2021. The results of the training model prove that the accuracy score is 69.5%, recall is 69.5%, and precision is 68.7%.</i></p>

I. PENDAHULUAN

Pemberlakuan Pembatasan Kegiatan Masyarakat atau yang biasa disebut PPKM merupakan salah satu kebijakan pemerintah untuk menanggulangi pandemi covid-19 yang tidak kunjung berhenti di Indonesia. PPKM yang berlaku pada saat penelitian ini berlangsung yaitu PPKM bertingkat dari level 1 - level 4. Pada mulanya PPKM bertingkat hanya diberlakukan di Jawa dan Bali (Krisdiyanto & Nurharyanto, 2021), namun dikarenakan di daerah lain mengalami peningkatan kasus yang signifikan maka PPKM diperluas ke 15 daerah yang lain. Pembatasan yang dilakukan meliputi *Work From Home* (bekerja dari rumah), sekolah daring, jam operasional di tempat keramaian, membatasi makan di tempat, dan sebagainya.

Sebelum PPKM bertingkat ini diberlakukan, kebijakan ini sebelumnya bernama PPKM Darurat, PPKM Mikro, dan PSBB (Krisdiyanto & Nurharyanto, 2021). Dengan kebijakan tersebut, banyak masyarakat yang geram dan mulai protes terhadap pemerintah terhadap kebijakan pemerintah yang belum matang karena tidak memikirkan masyarakat di kalangan bawah yang memiliki penghasilan harian (Wati & Ernawati, 2021). Banyaknya video pelaku UMKM yang protes terhadap kebijakan PPKM tersebar di sosial media menjadi alasan yang kuat untuk melakukan penelitian ini. Namun, tidak sedikit pula orang yang mendukung PPKM diperpanjang dengan berbagai alasannya masing-masing seperti mengikuti anjuran pemerintah, kesehatan, lebih menyukai sekolah daring, dan sebagainya (Putra et al., 2021).

Sosial media yang paling populer untuk mengutarakan pendapat adalah Twitter (Syarifuddin, 2020). Twitter merupakan sosial media yang lebih memprioritaskan bersosial menggunakan teks meskipun pada versi yang baru telah mendukung format video dan foto sebagai pendukung cuitan. Dengan begitu twitter merupakan sarana yang tepat untuk pengambilan data sentimen masyarakat di internet. Selain itu Twitter juga sedang naik daun sesaat sebelum pandemi berlangsung dan melambung tinggi saat pandemi berlangsung (Sihombing & Nataliani, 2021).

Dikarenakan banyaknya pendapat masyarakat yang bermacam-macam mengenai Perpanjangan PPKM, maka dari itu dibutuhkan suatu program yang dapat mengklasifikasikan positif atau negatif sentimen masyarakat (Samsir et al., 2021) terhadap perpanjangan PPKM yang dapat digunakan disetiap hari pengumuman perpanjangan PPKM. Oleh karena itu peneliti

mengusulkan untuk membuat model klasifikasi menggunakan algoritma k-nearest neighbor.

Penelitian analisis sentimen pengguna twitter tentang topik Pilkada DKI 2017 dilakukan oleh (Deviyanto & Wahyudi, 2018) dengan menerapkan algoritma KNN (*K-Nearest Neighbor*) sebagai metode penelitiannya. Data yang digunakan pada penelitian ini merupakan data tweet berbahasa Indonesia yang telah dikumpulkan selama bulan Januari 2017 dengan menggunakan library Python yaitu *Twitterscraper*. Pengklasifikasian nilai sentimen positif dan negatif dilakukan dengan bantuan pembobotan kata TF-IDF dan fungsi Cosine Similarity. Nilai akurasi terbesar pada hasil pengujian yaitu diketahui sebesar 67,2% dengan nilai $k = 5$, sedangkan untuk presisi tertinggi yaitu 56,94% dengan $k=5$, dan nilai recall tertinggi yaitu 78,24% dengan $k=15$. Perbedaan penelitian ini dengan penelitian yang dilakukan oleh (Deviyanto & Wahyudi, 2018) yaitu pada topik yang diteliti, pada penelitian ini topik yang diangkat yaitu tentang "Perpanjangan PPKM".

Penelitian yang serupa dengan penelitian ini yaitu penelitian yang dilakukan oleh (Syarifuddin, 2020). Topik yang diangkat oleh Syarifuddin merupakan kebijakan sebelum PPKM yaitu Opini Publik mengenai PSBB pada Twitter. Algoritma yang digunakan pada penelitian ini juga merupakan salah satu algoritma yang digunakan oleh Syarifuddin. Syarifuddin menggunakan 3 algoritma yaitu *Decision Tree*, KNN, dan Naïve Bayes. Syarifuddin menggunakan 3 algoritma dengan tujuan untuk mencari nilai akurasi yang terbaik dalam proses prediksi. Dari hasil ketiga algoritma yang digunakan, hasil terbaik dilakukan oleh algoritma *Decision Tree* dengan perolehan nilai akurasi 83,3%, nilai presisi 79% dan recall 87,17%.

Penelitian yang dilakukan oleh (Septian et al., 2019) berbeda dengan kedua peneliti sebelumnya, Septian dan rekannya tidak meneliti suatu kasus yang berhubungan dengan pemerintahan melainkan Polemik Persepakbolaan Indonesia dengan kata "@pssi" sebagai kata kunci dalam pengumpulan datanya melalui situs Twitter dengan library *Tweepy* sebagai alatnya. Klasifikasi menggunakan algoritma *K-Nearest Neighbor* dengan 2000 data tweets yang digunakan membuahkan hasil sangat baik yaitu dengan $k=23$ memperoleh hasil akurasi 79,9%.

Masih seputar analisis sentimen terhadap pengguna twitter, (Mahfud et al., 2020) melakukan penelitian mengenai Perpustakaan Nasional Republik Indonesia. Tujuan dilakukannya penelitian ini merupakan untuk mengukur tingkat kepuasan pengunjung

terhadap fasilitas dan layanan dari Perpustakaan Nasional Republik Indonesia. Data yang dikumpulkan yaitu dari Januari 2019 sampai dengan Agustus 2019 memperoleh 522 data dengan 72 tweets positif, 18 tweets negatif, dan 433 tweets netral. Untuk proses klasifikasi yang dilakukan, algoritma yang digunakan oleh Mafud dan rekannya yaitu algoritma Naïve Bayes dan K-Nearest Neighbor. Dari dua algoritma berikut, hasil terbaik dihasilkan oleh algoritma *K-Nearest Neighbor* dengan nilai akurasi 83.33%, nilai presisi 79.2%, dan nilai recall 83.3%. Hasil yang sangat memuaskan karena ketiga nilai ada dikisaran 80%.

Topik yang diangkat oleh (Lestari & Mahdiana, 2021) merupakan imbas dari kebijakan PPKM yaitu Sentimen Masyarakat terhadap Larangan Mudik 2021. Data yang digunakan pada penelitian berasal dari cuitan pengguna twitter dengan jumlah data sebesar 4.799 data tweet yang diambil dari 4 April 2021 sampai dengan 17 Mei 2021. Sentimen dari data tersebut yaitu 834 tweet positif dan 3.965 tweet negatif. Proses klasifikasi dilakukan menggunakan algoritma KNN dengan $k=3$ memperoleh nilai akurasi sebesar 86.67%, nilai recall 39.52, nilai presisi 70.97%, dan spencificity sebesar 96.60%. Berdasarkan penelitian tersebut peneliti menyimpulkan bahwa masyarakat keberatan dengan kebijakan larangan mudik tersebut.

II. BAHAN DAN METODE

A. Pelabelan Data

Data yang berhasil dikumpulkan menggunakan library Scweet, data yang diperoleh diberi label sentimen sebagai syarat untuk melakukan tahapan klasifikasi. Pemberian label dilakukan secara manual oleh tenaga manusia yaitu peneliti. Pemberian label dilakukan oleh tenaga manusia dengan asumsi hanya makhluk hidup yang dapat merasakan emosi atau sentimen, sedangkan mesin tidak demikian. Data yang diberi label yaitu data pada bulan Juli, data bulan Juli akan digunakan sebagai data train saat proses training model dengan algoritma K-NN. Sedangkan data bulan Oktober sampai dengan bulan Desember akan diberi label dengan cara prediksi menggunakan model K-NN yang telah ditraining. Label data yang diberikan yaitu 0, 1, dan 2. Label 0 berarti negatif, label 1 berarti netral, sedangkan label 2 berarti positif. Data negatif yang dimaksud yaitu data yang berisi mengenai cemoohan, kata-kata kasar, protes terhadap perpanjangan PPKM, dan menolak keras perpanjangan PPKM. Sedangkan

data netral yaitu data tweet yang berisikan berita, promosi, dan hal-hal lain yang tidak ada unsur sentimen dalam penetikannya. Data positif berisi mengenai dukungan masyarakat terhadap kebijakan PPKM, tweet yang bersifat mengingatkan tentang kebijakan PPKM, tweet yang saling mendoakan.

B. Data Preprocessing

Setelah data sudah selesai dilabeli, maka data harus melalui tahap data preprocessing sebelum data digunakan untuk training model. Data preprocessing yang dilakukan antara lain Data Cleansing, Normalisasi, Tokenizing, Remove Stopwords, dan Weighting. Perintah-perintah yang digunakan untuk proses preprocessing dibuat sebagai suatu function agar mempermudah dan mempercepat dalam penerapan ke data-data yang digunakan.

1. Data Cleansing

Proses pembersihan data dilakukan menggunakan library pandas untuk membersihkan data duplikat. Setelah dilakukan proses pembersihan data duplikat, data yang berhasil dibersihkan yaitu sebanyak 0 data yang berarti sejak awal data tidak ada yang duplikat.

2. Normalisasi

Proses Normalisasi dilakukan sebanyak 2 kali. Yang pertama dilakukan untuk membersihkan simbol-simbol, baris, angka, emoji, link, hashtag, dan mention. Tahap normalisasi pertama dilakukan menggunakan library regex dengan perintah seperti berikut:

```
re.sub("(@[A-Za-z0-9_+])|(\\d)|#[0-9A-Za-z_+]|([^\A-Za-z \t])|(\\w+:\\\\|\\S+)",  
",text)
```

Penjelasan:

re.sub: berfungsi untuk me-replace suatu teks dengan kriteria/kondisi tertentu.

Kriteria dalam perintah tersebut terdiri dari 5 grup kriteria :

- @[A-Za-z0-9_+] : untuk menyeleksi retweet, cara kerjanya yaitu dengan mencari kata yang diawali tanda "@".
- (\\d) : untuk menyeleksi angka desimal.
- #[0-9A-Za-z_+] : untuk menyeleksi tagar, cara kerjanya yaitu dengan mencari kata yang diawali tanda "#"
- ([^\A-Za-z \t]) : untuk menyeleksi emoji dan tanda baca, cara kerjanya dengan menyeleksi selain huruf alfabet, spasi, dan tab spasi.
- (\\w+:\\\\|\\S+) : untuk menyeleksi link pada teks, cara kerjanya yaitu mencari kata yang disambung dengan simbol

“:./” dengan terusan karakter yang bukan tergolong dalam whitespace (seperti spasi, tab spasi, baris baru, dll).

Tahap normalisasi kedua yaitu melakukan lowercase terhadap data yang telah dibersihkan. Tahap normalisasi ini bertujuan untuk menyamakan term yang dibedakan oleh huruf kecil dan huruf kapital karena algoritma bersifat case sensitive. Proses ini tidak menggunakan bantuan library manapun namun menggunakan perintah yang sudah terdapat pada bahasa pemrograman Python.

3. *Tokenizing*

Tahap tokenizing atau tahap memisahkan kata dilakukan menggunakan bantuan library NLTK. Kalimat dipisahkan dengan tujuan untuk digunakan pada proses pembobotan TF-IDF.

4. *Remove Stopwords*

Pada tahap penghapusan stopwords atau juga disebut sebagai filtering dilakukan menggunakan bantuan library NLTK. Penghapusan stopwords dilakukan untuk menghapus kata-kata yang tidak bermakna, pada library NLTK terdapat fitur penghapusan stopwords Bahasa Indonesia. Namun ada kekurangan jika hanya mengandalkan stopwords default yang terdapat pada suatu library. Karena kata-kata yang disingkat atau typo tidak akan terdeteksi, dengan demikian kata-kata yang typo dan disingkat akan dianggap sebagai suatu kata/term baru meskipun maknanya sama.

5. *Weighting*

Weighting atau dalam Bahasa Indonesia berarti pembobotan, dilakukan untuk membedakan kata (*term*) yang sering muncul dan jarang muncul. Pembobotan dilakukan dengan metode TF-IDF (*Term Frequency - Inverse Document Frequency*). Pada penerapannya dilakukan menggunakan bantuan library Scikit-Learn dan diterapkan pada pipeline karena *pipeline* juga merupakan bagian dari *library Scikit-Learn*.

Preprocessing Data Cleansing, Normalisasi, *Tokenizing*, dan *Remove Stopwords* akan digabungkan dalam satu *function* agar mempercepat dalam proses prediksi pada bulan-bulan berikutnya. Perintah *function* yang dibuat sebagai berikut :

```
def proses(df):
```

```
df['tweet'] = df['tweet'].apply(bersih)
df['tweet'] = df['tweet'].apply(lowertext)
df['tweet'] = df['tweet'].apply(tokenizingText)
df['tweet'] = df['tweet'].apply(filteringText)
df['tweet'] = df['tweet'].apply(toSentence)
return df
```

Perintah tersebut diterapkan pada setiap *data frame*. Setelah melewati tahapan *Data Cleansing*, *Tokenizing*, dan *Remove Stopwords* maka akan disatukan kembali menjadi satu kalimat. Karena TF-IDF yang ada pada Scikit-learn membutuhkan suatu kalimat utuh, bukan terpisah (*tokenized*). Perintah penyatuan kalimat (*toSentence*) sebagai berikut :

```
def toSentence(list_words):
    sentence = ''.join(word for word in list_words)
    return sentence
```

C. *Data Splitting*

Setelah data diolah, tahap selanjutnya yaitu *data splitting*. Data akan dibagi menjadi 2 bagian yaitu data train dan data test. *Data splitting* dilakukan dengan rasio 8:2 yang berarti 80% data train dan 20% data test. Setelah dilakukan *data splitting* diperoleh 916 data train dan 230 data test. Proses ini dilakukan dengan bantuan library Scikit-learn. Perintah yang dijalankan yaitu sebagai berikut :

```
X = df.tweet
y = df.Label
X_train, X_test, y_train, y_test = train_test_split(X,
    y, test_size=0.2, random_state=23, stratify=y)
```

Penjelasan :

X : teks data tweet

y : label dari data tweet

test_size : ukuran data test yang diinginkan

random_state : digunakan untuk mengunci keacakan, karena setiap kali dilakukan *splitting* data akan diacak.

stratify : rasio data label antara data test dan data train akan dibagi secara merata.

D. *Tuning*

Parameter yang digunakan untuk tuning model yaitu jumlah k, bobot, dan rumus jarak. Jumlah k yang digunakan yaitu angka ganjil dari 3 sampai 21 agar hasil prediksi stabil karena dalam penentuan label K-NN melakukan voting data yang terdekat. Bobot yang digunakan untuk tuning yaitu *uniform* dan *distance*. Perbedaannya yaitu pada pembobotan *uniform* K-NN hanya akan melakukan voting jumlah data terdekat tanpa memperdulikan jarak, sedangkan pada pembobotan *distance* jarak antar label terdekat akan dijumlah dan pemenang hasil votingnya yaitu label dengan jumlah jarak terdekat. Rumus jarak yang digunakan untuk *tuning parameter*

yaitu *manhattan distance* dan *euclidean distance* dengan *metric minkowski*. Parameter yang telah ditentukan disimpan kedalam variabel “params” untuk dipanggil saat proses *training*. Berikut perintah yang digunakan :

```
params = {  
    'algo_n_neighbors': range(3,21,2),  
    'algo_weights': ['uniform', 'distance'],  
    'algo_p': [1, 2]  
}
```

Pada penamaan parameter terdapat “algo_” dilakukan karena algoritma K-NN disimpan dalam pipeline “algo”.

E. Training

Pada fase training kita menggunakan pipeline dari Scikit-learn dan juga GridSearchCV dari Scikit-learn untuk tuning dan training model. Pipeline digunakan untuk mempermudah dalam proses fitting pada data yang dipreproses dengan algoritma K-NN. Sedangkan GridSearchCV digunakan untuk tuning model / mencari model yang terbaik dari beberapa parameter menggunakan metode Cross Validation. Berikut merupakan perintah-perintah yang digunakan saat training :

```
pipeline = Pipeline([  
    ('prep', TfidfVectorizer()),  
    ('algo', KNeighborsClassifier())  
])
```

Proses pembuatan pipeline dengan ketentuan preprocessor TF-IDF dan algoritma K-NN. Preprocessor disini hanya bisa diisi menggunakan preprocessor yang tersedia pada library Scikit-learn.

```
model = GridSearchCV(pipeline, params, cv=3,  
n_jobs=-1)  
model.fit(X_train, y_train)
```

Proses training dan fitting dilakukan dengan GridSearchCV. Pipeline menunjukkan estimator/algoritma kita yang berwujud pipeline, params menunjukkan parameter-parameter yang telah kita buat, cv=3 menunjukkan jumlah cross validation yang digunakan, n_jobs=-1 menunjukkan bahwa kita menggunakan seluruh core processor dalam proses training untuk mempercepat proses training. Setelah pembuatan GridSearchCV, maka dilakukan fitting pada data X_train dan y_train.

```
print(model.best_params_)
```

Perintah ini digunakan untuk mencetak model dengan parameter terbaik berdasarkan hasil GridSearchCV. Didapatkan hasil sebagai berikut :

```
{'algo_n_neighbors': 13, 'algo_p': 2,  
'algo_weights': 'distance'}
```

Parameter terbaik yang didapatkan yaitu jumlah tetangga (k) sebanyak 13, rumus yang

digunakan yaitu euclidean distance, dan pembobotan jarak yang digunakan yaitu distance.

```
print(model.score(X_train, y_train),  
model.best_score_, model.score(X_test, y_test))
```

Perintah di atas digunakan untuk mencetak skor model yang diperoleh pada data train, skor terbaik saat validasi, dan juga skor pada data test. Hasil yang diperoleh dapat dilihat pada gambar 3.13. Skor pada data train yaitu 0.998, skor saat validasi yaitu 0.648, skor pada data test yaitu 0.695. Ini menandakan bahwa model sangat akurat saat fase training, namun setelah dites dengan data tes hasilnya tidak begitu akurat.

Testing

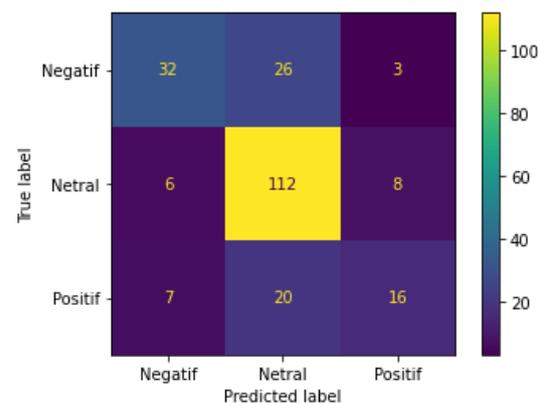
Setelah melakukan training, selanjutnya melakukan tes prediksi pada data test, lalu hasil prediksi dibandingkan dengan hasil yang sebenarnya menggunakan confusion matrix dengan perintah-perintah berikut:

```
y_pred = model.predict(X_test)
```

Melakukan prediksi terhadap data X_test dan disimpan dalam variabel y_pred. Setelah dilakukan prediksi, maka akan dilakukan visualisasi confusion matrix agar mudah dimengerti.

```
cm = confusion_matrix(y_test, y_pred,  
labels=model.classes_)  
disp = ConfusionMatrixDisplay(confusion_matrix=cm,  
display_labels=['Negatif', 'Netral', 'Positif'])  
disp.plot()
```

Perhitungan confusion matrix dilakukan menggunakan bantuan Scikit-learn. Setelah dilakukan prediksi, hasil prediksi yang disimpan pada variabel y_pred akan dibandingkan dengan hasil yang sesungguhnya (y_test). Untuk memudahkan dalam pembacaan matrix, label 0, 1, dan 2 yang terdapat pada model.classes_ diganti menjadi Negatif, Netral dan Positif. Hasil perintah diatas dapat dilihat pada gambar 11 :



Sumber: Hasil Penelitian (2022)

Gambar 1. Confusion Matrix

1. Label Negatif yang benar diprediksi : 32 data
2. Label Negatif yang diprediksi Netral : 26 data
3. Label Negatif yang diprediksi Positif : 3 data
4. Label Netral yang diprediksi Negatif : 6 data
5. Label Netral yang benar diprediksi : 112 data
6. Label Netral yang diprediksi Positif : 8 data
7. Label Positif yang diprediksi Negatif : 7 data
8. Label Positif yang diprediksi Netral : 20 data
9. Label Positif yang benar diprediksi : 16 data

Berdasarkan poin-poin diatas maka dapat disimpulkan bahwa Prediksi yang paling akurat yaitu pada label Netral. Sedangkan yang terburuk merupakan pada label Positif karena prediksi yang benar lebih sedikit dari prediksi yang salah.

Setelah itu dilakukan perhitungan skor dengan menggunakan bantuan Scikit-learn. Perhitungan yang dihasilkan sebagai berikut :

Accuracy Score : 0.6956521739130435
Recall Score : 0.6956521739130435
Precision Score : 0.6877204998063556

Sumber: Hasil Penelitian (2022)

Gambar 2. Skor Model

Hasilnya tidak terlalu meyakinkan karena skor hanya mendekati 70%. Artinya akan ada sekitar 30% prediksi yang salah.

Predict

Pada tahap ini, dilakukan prediksi terhadap data pada bulan Agustus sampai dengan bulan Desember 2021. Proses yang dilalui yaitu data akan dipreproses seperti data train lalu akan diprediksi labelnya. Perintah yang digunakan untuk memprediksi sebagai berikut :

```
proses(dtAgustus)
dtAgustus['label'] =
model.predict(dtAgustus.tweet)
```

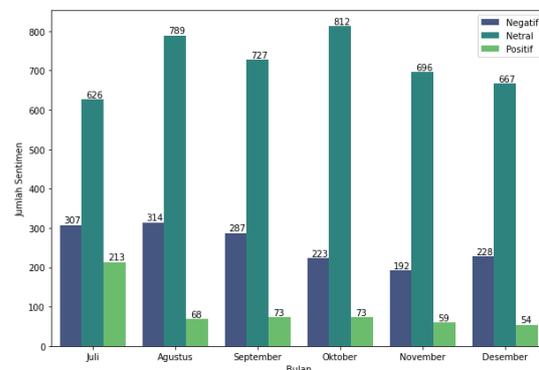
Perintah tersebut merupakan perintah untuk prediksi data bulan Agustus dan hasil prediksi disimpan pada kolom "label". Perintah diatas diterapkan pada bulan-bulan selanjutnya sampai dengan bulan Desember.

Visualisasi

Visualisasi yang dilakukan meliputi sentimen yang terdapat pada tiap bulan, persentase sentimen secara keseluruhan (bulan Juli - Desember), pertumbuhan sentimen dari

bulan ke bulan, dan sebuah wordcloud dari data yang dikumpulkan.

1. Visualisasi Sentimen pada Setiap Bulan
Visualisasi pertama yaitu dilakukan untuk memvisualisasikan jumlah sentimen yang ada pada tiap bulan dari bulan Juli sampai dengan bulan Desember. Hasil yang didapatkan dapat dilihat pada gambar dibawah ini :



Sumber: Hasil Penelitian (2022)

Gambar 3. Visualisasi Jumlah Sentimen per Bulan

Angka sentimen Positif pada bulan Juli paling banyak dikarenakan data yang diperoleh lebih banyak daripada bulan-bulan yang lain. Tetapi, secara persentase dari bulan Juli - bulan Desember sentimen Negatif lebih dominan dibandingkan dengan sentimen Positif. Hal ini menunjukkan bahwa sentimen pengguna twitter terhadap PPKM yang diperpanjang bersentimen negatif (tidak senang dengan kebijakan tersebut).

2. Visualisasi Persentase Sentimen

Visualisasi dilakukan untuk melihat dengan sudut pandang yang luas, seperti apa sentimen cuitan masyarakat di media sosial Twitter. Salah satu cara untuk mengetahuinya yaitu dengan melihat persentase sentimen yang ada pada seluruh bulan dari Bulan Juli sampai dengan Bulan Desember. Gambar 4.7 menunjukkan persentase sentimen masyarakat pada media sosial Twitter terhadap kata kunci "PPKM" :

Dengan menggunakan Algoritma K-NN yang dilengkapi dengan GridSearchCV, didapatkan skor akurasi sebesar 69,5%, recall 69,5%, dan presisi 68,7%. Hasil yang kurang memuaskan karena kita membutuhkan akurasi yang tinggi agar valid jika digunakan untuk evaluasi kebijakan.

REFERENSI

- Deviyanto, A., & Wahyudi, M. D. R. (2018). Penerapan Analisis Sentimen Pada Pengguna Twitter Menggunakan Metode K-Nearest Neighbor. *JISKA (Jurnal Informatika Sunan Kalijaga)*, 3(1), 1-13. <https://doi.org/10.14421/jiska.2018.31-01>
- Krisdiyanto, T., & Nurharyanto, E. M. O. (2021). Analisis Sentimen Opini Masyarakat Indonesia Terhadap Kebijakan PPKM pada Media Sosial Twitter Menggunakan Naïve Bayes Clasifiers. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer Dan Teknologi Informasi*, 7(1), 32-37. <http://ejournal.uin-suska.ac.id/index.php/coreit/article/view/12945>
- Lestari, D. A., & Mahdiana, D. (2021). Penerapan Algoritma K-Nearest Neighbor pada Twitter untuk Analisis Sentimen Masyarakat Terhadap Larangan Mudik 2021. *JURNAL INFORMATIK*, 17(2), 123-131. <https://ejournal.upnvj.ac.id/index.php/informatik/article/view/3629/1405>
- Mahfud, F. K. R., Mudawamah, N. S., & Hariyanto, W. (2020). Sentiment Analysis of Perpustakaan Nasional Republik Indonesia Through Social Media Twitter. *Matics*, 12(1), 90-93. <https://doi.org/10.18860/mat.v12i1.8973>
- Putra, A., Haeirudin, D., & Khairunnisa, H. (2021). Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Pada Media Sosial Twitter Menggunakan Algoritma Svm. *November*.
- Samsir, Ambiyar, Verawardina, U., Edi, F., & Watianthos, R. (2021). Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 Menggunakan Metode Naïve Bayes. *Jurnal Media Informatika Budidarma*, 5(1), 157-163. <https://doi.org/10.30865/mib.v5i1.2604>
- Septian, J. A., Fahrudin, T. M., & Nugroho, A. (2019). Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor. *JOURNAL OF INTELLIGENT SYSTEMS AND COMPUTATION*, 43-49. <https://t.co/9WloaWpfd5>
- Sihombing, D. Y., & Nataliani, Y. (2021). Analisis Interaksi Pengguna Twitter pada Strategi Pengadaan Barang Menggunakan Social Network Analysis. *Sistemasi: Jurnal Sistem Informasi*, 10(2), 434-444. <https://doi.org/10.32520/stmsi.v10i2.1289>
- Syarifuddin, M. (2020). Analisis Sentimen Opini Publik Terhadap Efek Psbb Pada Twitter Dengan Algoritma Decision Tree-Knn-Naïve Bayes. *INTI Nusa Mandiri*, 15(1), 87-94. <https://doi.org/10.33480/inti.v15i1.1433>
- Wati, R., & Ernawati, S. (2021). Analisis Sentimen Persepsi Publik Mengenai PPKM Pada Twitter Berbasis SVM Menggunakan Python. *Jurnal Teknik Informatika UNIKA Santo Thomas*, 06, 240-247. <http://ejournal.ust.ac.id/index.php/JTIUST/article/view/1465>