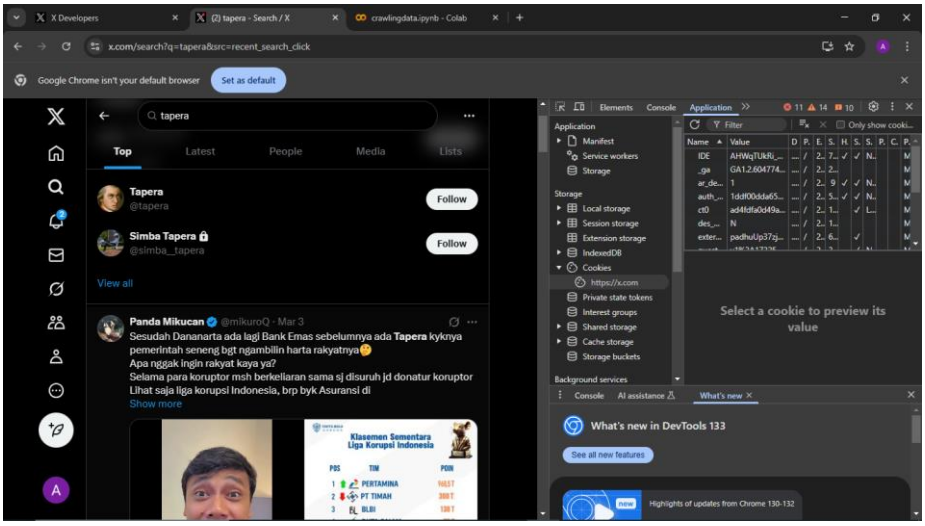
# BAB IV

# HASIL DAN PEMBAHASAN

Bab ini akan menyajikan hasil analisis perbandingan sentimen terhadap tapera diplatform media sosial twitter menggunakan dua metode, yaitu K-Nearst Neighbors (KNN) dan Support Vector Machine (SVM). Tujuannya adalah mengidentifikasi polaritas sentimen publik terhadap tapera dan mengevaluasi akurasi kedua metode dalam klasifikasi sentimen.

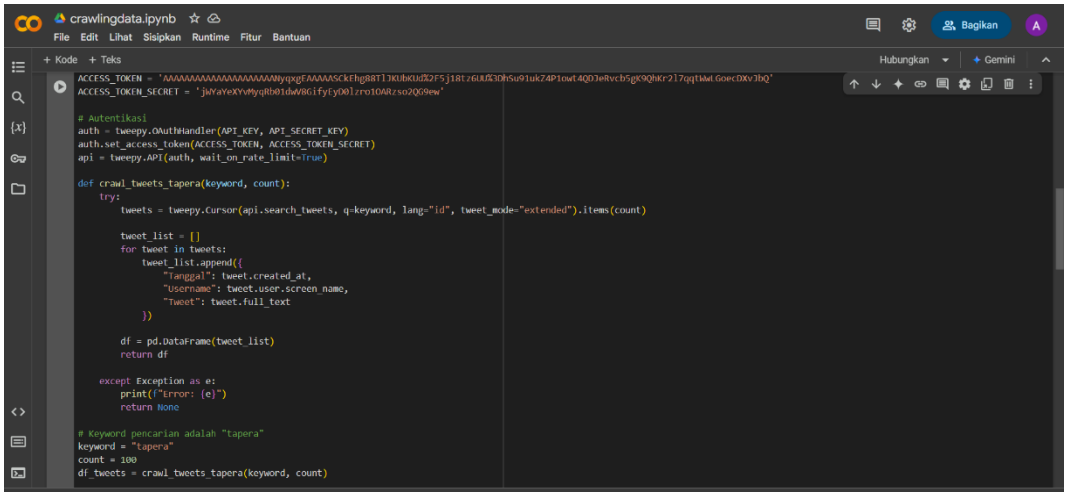
## 4.1 Proses pengumpulan data

Tahap ini merupakan tahap awal dimana melakukan pengumpulan *dataset* yang akan digunakan pada proses selanjutnya. *Dataset* yang digunakan berupa tweet atau unggahan pengguna media sosial twitter/X.



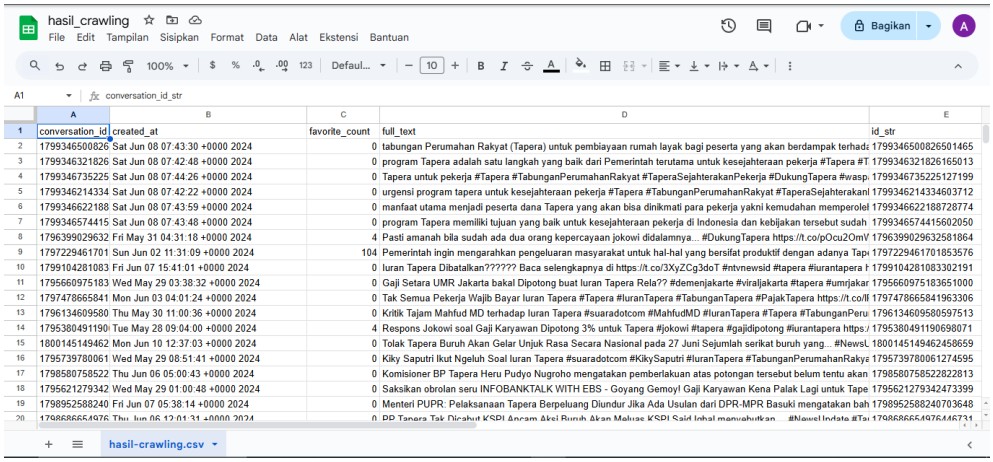

**Gambar 4. 1 mengambil data menggunakan api**

Pada Gambar 4.1 merupakan pengambilan API *key* pada media sosial twitter/x dengan mengambil data *tweet* nya yang akan digunakan pada metode crawling. Keyword yang digunakan pada data *tweet* media sosial twitter/x yaitu #Tapera.




**Gambar 4. 2 proses crawling menggunakan google colab**

Pada Gambar 4.2 merupakan pengumpulan data menggunakan metode *crawling* dengan *application interface* (API) *key* pada media sosial x dengan mengambil *tweet* menggunakan bahasa pemrograman python. Data yang diperoleh berupa teks dengan jumlah data 2366 dan 15 variabel terdiri dari *converstation_id*, *created_at*, *favorite_count*, *full_text*, *id_str*, *image_url*, *in_replay_to_screen_name*, *lang*, *location*, *quote_count*, *reply_count*, *retweet_count*, *tweet_url*, *user_id_str*, *username***.** Hasil dapat dilihat pada Gambar 4.3



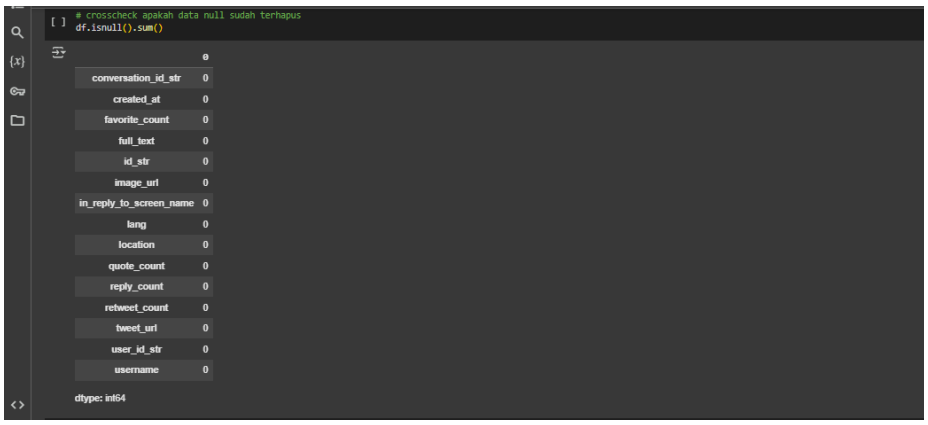

**Gambar 4. 3 data hasil crawling**

## 4.2 Proses pengecekan Missing value

```
# melihat jumlah data yang missing
data.isnull().sum()
```

**Gambar 4. 4 code proses missing value**

Pada tahap missing value yaitu memeriksa dan menghitung jumlah nilai hilang (missing values) di setiap kolom dalam dataset agar data menjadi lengkap dan siap dianalisis. Informasi mengenai jumlah dan persentase missing values sangat penting untuk menentukan strategi yang tepat untuk menanganinya. Dengan penanganan yang tepat, kita dapat memastikan bahwa data yang digunakan dalam analisis selanjutnya memiliki kualitas yang baik dan menghasilkan hasil yang akurat.

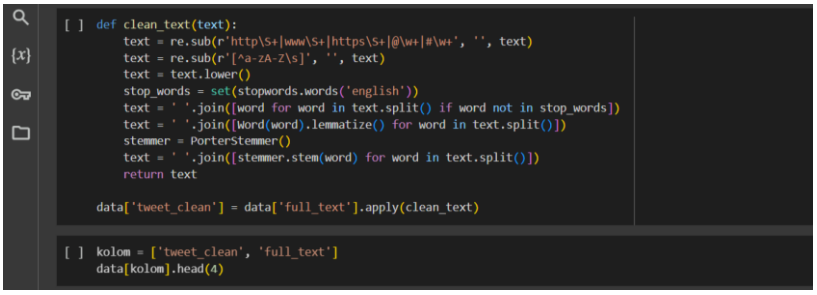



**Gambar 4. 5 missing value**

## 4.3 Prepocessing

Pada bagian preposesing data menjelaskan proses tahapan mengelola dan mempersiapkan data yang akan digunakan dari data mentah. Sebelum dilakukan tahap cleaning data mentah akan di drop pada bagian variabel yang tidak digunakan agar mempermudah proses dengan menggunakan data yang diperlukan agar data menjadi siap dianalisis. Pada tahap ini dilakukan pengurangan variabel dari 15 variabel menjadi 11 variabel selanjutnya baru dilakukan proses cleaning.

### 4.3.1 cleaning



```
[ ] def clean_text(text):
        text = re.sub(r'http\S+|www\S+|https\S+|@\w+|#\w+', '', text)
        text = re.sub(r'[^a-zA-Z\s]', '', text)
        text = text.lower()
        stop_words = set(stopwords.words('english'))
        text = ' '.join([word for word in text.split() if word not in stop_words])
        text = ' '.join([Word(word).lemmatize() for word in text.split()])
        stemmer = PorterStemmer()
        text = ' '.join([stemmer.stem(word) for word in text.split()])
        return text

    data['tweet_clean'] = data['full_text'].apply(clean_text)

[ ] kolom = ['tweet_clean', 'full_text']
    data[kolom].head(4)
```

**Gambar 4. 6 code dari cleaning**

Setelah data ulasan berhasil dikumpulkan perlu dilakukan pemrosesan data agar dapat menganalisa ulasan yang lebih akurat dengan menggunakan cleansing data berupa penghapusan karakter khusus, seperti spasi yang berturut-turut menggunakan google colab. Dengan hasil pembeda sebagai berikut:

**Table 2 hasil cleaning**

| Data Tweet | Hasil Cleaning |
|---|---|
| tabungan Perumahan Rakyat (Tapera) untuk pembiayaan rumah layak bagi peserta yang akan berdampak terhadap kesejahteraan pekerja. #Tapera #TabunganPerumahanRakyat #TaperaSejahterakanPekerja #DukungTapera #waspadaiprovokasi https://t.co/GgZDJsUICo | tabungan perumahan rakyat tapera untuk pembiayaan rumah layak bagi peserta yang akan berdampak terhadap kesejahteraan pekerja |
| Berat! Di Zaman Distrust Gaji Karyawan Kena Palak untuk Tapera Apa Urgensinya di Tengah Beban Hidup Karyawan yang Berat? Selengkapnya: https://t.co/2eg73SMbs9 #infobank #bank #banking #perbankan #tapera | berat di zaman distrust gaji karyawan kena palak untuk tapera apa urgensinya di tengah beban hidup karyawan yang berat selengkapnya |

| | |
|---|---|
| #tabunganperumahanrakyat #karyawan #gaji #potongangaji #iurantapera https://t.co/WfEoubxYpo | |
| Gaji PNS-Karyawan Swasta Dipotong 2 5% untuk Iuran Tapera Berikut Mekanisme & Manfaatnya https://t.co/E7wbicFcOM #tapera #iurantapera | gaji pns karyawan swasta dipotong untuk iuran tapera berikut mekanisme & amp manfaatnya |
| Pemerintah ingin mengarahkan pengeluaran masyarakat untuk hal-hal yang bersifat produktif dengan adanya Tapera. #MetroTVKontroversi #tapera #iurantapera #tabunganperumahanrakyat https://t.co/v0MWQmztS5 | pemerintah ingin mengarahkan pengeluaran masyarakat untuk halhal yang bersifat produktif dengan adanya tapera |
| Meskipun bertujuan membantu pekerja berpenghasilan rendah memiliki rumah kebijakan ini mendapat penolakan dari pengusaha dan pekerja swasta karena potongan gaji dan iuran tambahan. artikel lainnya di https://t.co/vwJXlFUaug #tapera #tabunganperumahanrakyat #potongantapera https://t.co/dUNTOEzSZd | meskipun bertujuan membantu pekerja berpenghasilan rendah memiliki rumah kebijakan ini mendapat penolakan dari pengusaha dan pekerja swasta karena potongan gaji dan iuran tambahan artikel lainnya di |

### 4.3.2 Tokenisasi

```
df['jumlah_char'] = df['tweet_clean'].str.len()
kolom = ['tweet_clean', 'jumlah_kata', 'jumlah_char']
df[kolom].head(4)
```

**Gambar 4. 7 Scrip tokenizing pada python**

Tokenisasi yaitu proses memecah teks menjadi unit yang lebih kecil seperti kata atau karakter. Berikut hasil yang telah dilakukan tokenisasi :

**Table 3 tokenizing**

| Tweet_clean | Jumlah_kata |
|---|---|
| tabungan perumahan rakyat tapera untuk pembiayaan rumah layak bagi peserta yang akan berdampak terhadap kesejahteraan pekerja | 16 |
| program tapera adalah satu langkah yang baik dari pemerintah terutama untuk kesejahteraan pekerja | 13 |
| tapera untuk pekerja | 3 |
| urgensi program tapera untuk kesejahteraan pekerja | 6 |
| manfaat utama menjadi peserta dana tapera yang akan bisa dinikmati para pekerja yakni kemudahan memperoleh rumah hingga adanya biaya untuk renovasi rumah | 22 |

### 4.3.3 Menghitung jumlah kata dalam teks

```
[ ] df['jumlah_kata'] = df['tweet_clean'].apply(lambda x: len(str(x).split(' ')))
    kolom = ['tweet_clean', 'jumlah_kata']
    df[kolom].head(4)
```

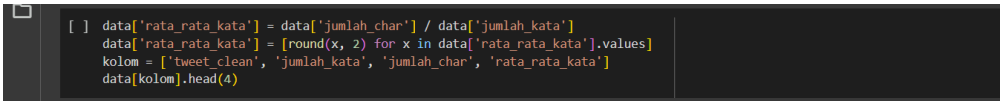
**Gambar 4. 8 Code menghitung jumlah kata**

Bertujuan untuk memberikan informasi tentang berapa banyak kata yang terkandung dalam setiap tweet. Berikut hasil dari menghitung jumlah kata:

**Table 4 hasil jumlah kata**

| Tweet_clean | Jumlah_kata | Jumlah _char |
|---|---|---|
| tabungan perumahan rakyat tapera untuk pembiayaan rumah layak bagi peserta yang akan berdampak terhadap kesejahteraan pekerja | 16 | 125 |
| program tapera adalah satu langkah yang baik dari pemerintah terutama untuk kesejahteraan pekerja | 13 | 97 |
| tapera untuk pekerja | 3 | 20 |
| urgensi program tapera untuk kesejahteraan pekerja | 6 | 50 |

### 4.3.3 menghitung rata-rata panjang kata




**Gambar 4. 9 code menghitung rata-rata panjang kata**

Proses perhitungan rata-rata panjang kata membantu mengidentifikasi tingkat kompleksitas teks berdasarkan ukuran rata-rata setiap kata. Berikut hasil dari rata-rata panjang kata:

**Table 5 menghitung rata rata panjang kata**

| Tweet_clean | Jumlah_kata | Jumlah_char | Rata_rata_kata |
|---|---|---|---|
| tabungan perumahan rakyat tapera untuk pembiayaan rumah layak bagi peserta yang akan berdampak terhadap kesejahteraan pekerja | 16 | 125 | 7,81 |
| program tapera adalah satu langkah yang baik dari pemerintah terutama untuk kesejahteraan pekerja | 13 | 97 | 7,46 |
| Tapera untuk pekerja | 3 | 20 | 6,67 |
| urgensi program tapera untuk kesejahteraan pekerja | 6 | 50 | 8,33 |

Berdasarkan Tabel  bahwa jumlah karakter dalam sebuah teks adalah 125 dan jumlah katanya 16, maka rata-rata panjang kata dihitung sebagai 125 dibagi 16 ,yaitu 7,81 karakter perkata.

## 4.4 Labeling

```
# Mengaplikasikan fungsi ke setiap tweet
data['sentiment'] = data['tweet_clean'].apply(vader_sentiment)
kolom = ['sentiment','tweet_clean', 'jumlah_kata', 'jumlah_char', 'rata_rata_kata']
data[kolom].head(10)
```

**Gambar 4. 10 perintah untuk menampilkan labeling sentimen**

Selanjutnya adalah proses labeling, labeling dilakukan untuk menentukan polaritas sentimen setiap tweet. Hasil label sentimen (positif,negatif atau netral) dari teks tweet yang telah dibersihkan. Hasil pelebelan ini kemudian disimpan dalam kolom sentimen.

**Table 6 hasil labeling**

| Labeling | Tweet_clean | Jumlah_kata | Jumlah_char | Rata_rata_kata |
|---|---|---|---|---|
| neutral | tabungan perumahan rakyat tapera untuk pembiayaan rumah layak bagi peserta yang akan berdampak terhadap kesejahteraan pekerja | 16 | 125 | 7,81 |
| neutral | program tapera adalah satu langkah yang baik | 13 | 97 | 7,46 |

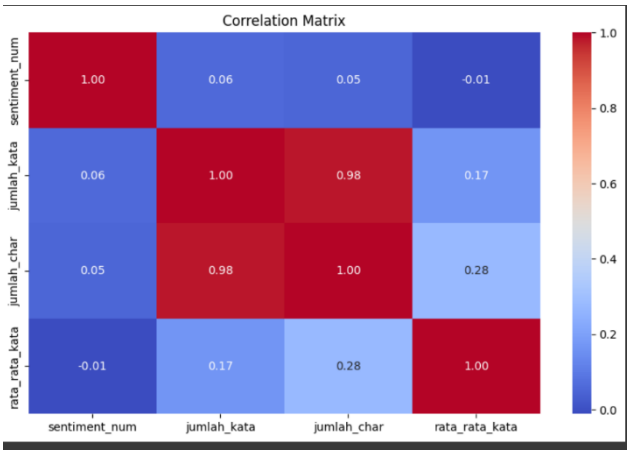
| | | | | |
|---|---|---|---|---|
| | dari pemerintah terutama untuk kesejahteraan pekerja | | | |
| neutral | tapera untuk pekerja | 3 | 20 | 6,67 |
| neutral | urgensi program tapera untuk kesejahteraan pekerja | 6 | 50 | 8,33 |
| neutral | manfaat utama menjadi peserta dana tapera yang akan bisa dinikmati para pekerja yakni kemudahan memperoleh rumah hingga adanya biaya untuk renovasi rumah | 22 | 153 | 6,95 |

## 4.5 Matrix colleration

```
# Visualisasi matriks korelasi
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
```

**Gambar 4. 11 visualisasi matrix colleration**

Mengukur dan menganalisis hubungan antar variabel dalam dataset. Matriks ini membantu mengidentifikasi seberapa kuat dan arah hubungan antar variabel (positif,negatif atau netral), sehingga dapat digunakan untuk pemilihan fitur,deteksi multikolinearitas, atau memahami pola data sebelum analisis lebih lanjut.




**Gambar 4. 12 matrix correlation**

Matriks korelasi di atas menunjukkan hubungan antar variabel:

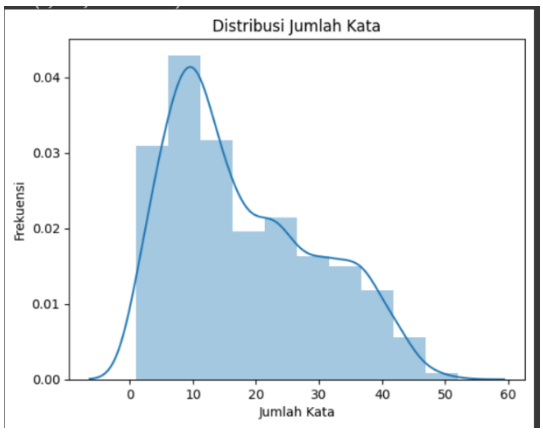
- jumlah_kata dan jumlah_char memiliki korelasi sangat kuat (0.98), artinya semakin banyak kata, semakin banyak karakter.
- Korelasi antara rata_rata_kata dan variabel lain relatif lemah (-0.01 hingga 0.28).
- Diagonal bernilai 1 karena itu hubungan variabel dengan dirinya sendiri.
- Warna merah menunjukkan korelasi positif tinggi, sedangkan biru menunjukkan korelasi negatif.

## 4.6 Analisis univariate jumlah data

```
# univariate analisis jumlah kata

# histogram distribusi jumlah kata
sns.distplot(data['jumlah_kata'], kde=True, bins=10)
plt.title('Distribusi Jumlah Kata')
plt.xlabel('Jumlah Kata')
plt.ylabel('Frekuensi')
```

**Gambar 4. 13 code histogram distribusi**

Analisis univariate menyajikan terhadap variabel “jumlah data” dalam dataset. Visualisasi histogram dan kernel density estimation (KDE) digunakan untuk menggambarkan distribusi data.




**Gambar 4. 14 analisis univariate jumlah data**

Grafik menunjukkan distribusi jumlah kata:

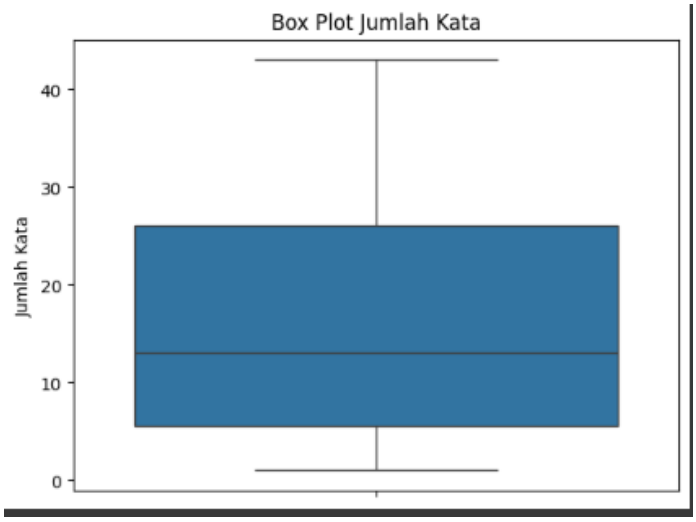
- Sumbu X: jumlah kata (0–60).
- Sumbu Y: frekuensi relatif.
- Mayoritas teks memiliki 5–20 kata, ditunjukkan oleh puncak histogram dan kurva biru.

## 4.7 Box plot

```
# box plot distribusi jumlah kata
sns.boxplot(data['jumlah_kata'])
plt.title('Box Plot Jumlah Kata')
plt.ylabel('Jumlah Kata')
```

**Gambar 4. 15 code visualisasi box plot**

Menyajikan visualisai distribusi jumlah kata dalam tweet melalui box plot. Box plot yang dihasilkan akan membantu untuk mengidentifikasi apakah ada outlier atau nilai ekstrem dalam data jumlah kata yang mungkin perlu penanganan khusus dalam analisis selanjutnya.




**Gambar 4. 16 box plot**

Gambar di atas merupakan Box Plot untuk distribusi jumlah kata. Cara membacanya:

- Kotak biru: menunjukkan rentang data dari kuartil pertama (Q1) hingga kuartil ketiga (Q3).
- Garis dalam kotak: menunjukkan median atau nilai tengah data.
- Garis horizontal di luar kotak: menunjukkan nilai minimum dan maksimum (tanpa outlier).

- Tidak ada outlier dalam grafik ini (titik di luar rentang tidak terlihat).

Kesimpulan: Sebagian besar jumlah kata berada di antara Q1 dan Q3, dengan nilai median mendekati 15 kata.

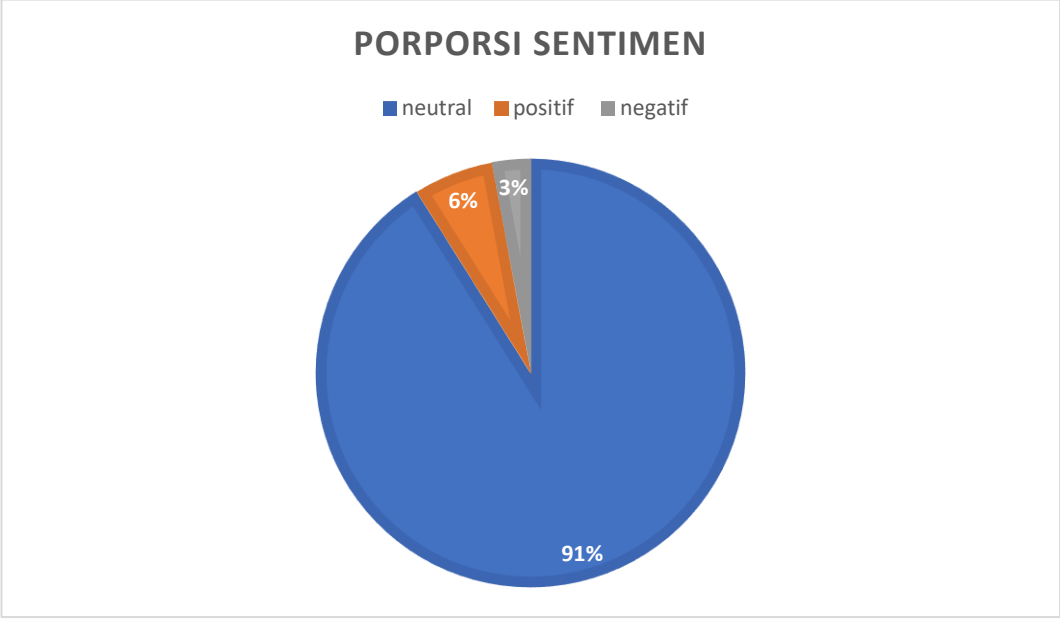
## 4.8 visualisasi data

### 4.9.1 perbandingan sentimen

```
# Menghitung jumlah masing-masing sentimen
sentimen_counts = data['sentiment'].value_counts()

# Plot diagram lingkaran
plt.figure(figsize=(6, 6))
plt.pie(sentimen_counts, labels=sentimen_counts.index, autopct='%1.1f%%', colors=['skyblue', 'lightgreen','lightcoral'])
plt.title('Proporsi Sentimen')
plt.show()
```

**Gambar 4. 17 code menghitung masing-masing sentimen**

Grafik 1 Hasil sentimen



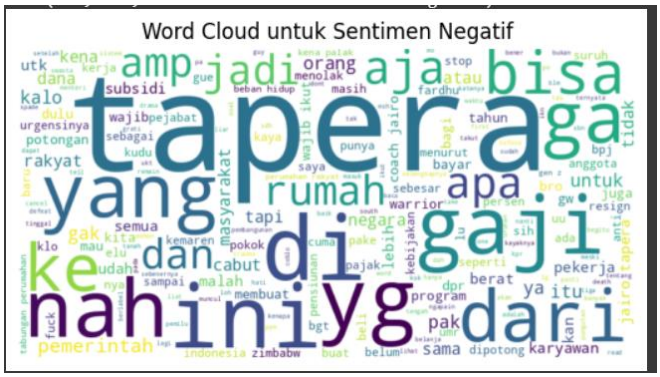

Hasil data dihitung menggunakan python yang kemudian data dilakukan visualisasi untuk mempermudah pembacaan hasil analisis sehingga menghasilkan pie chart seperti gambar diatas, Dengan mencoba memasuki kata kunci “Tapera” opini masyarakat tentang tapera yaitu sebanyak 91.0% netral, 6.0% positif dan 2.9% negatif. Dari hasil dapat ditarik kesimpulan bahwa python bisa mengklasifikasi tweet

dari twitter sehingga kita bisa mengidentifikasi opini masyarakat tentang tapera diindonesia yang akan mendatang dimana opininya netral.

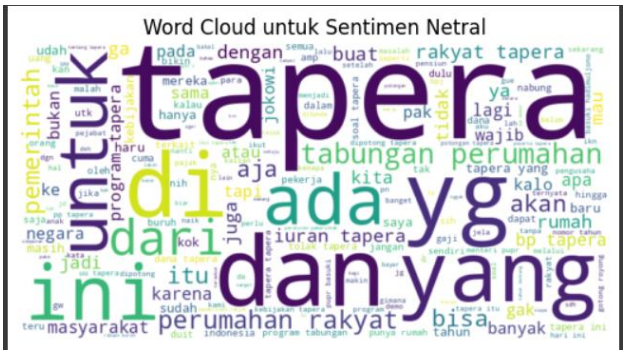
### 4.9.2 word cloud

word cloud berfungsi untuk menampilkan data text secara visual. Semakin besar kata tersebut maka jumlah frekuensi kata tersebut semakin banyak. Dapat diketahui bahwa jumlah kata yang paling sering muncul adalah tapera. Selain kata yang paling banyak muncul, word cloud juga bisa menampilkan kata positif,negatif dan netral yang paling banyak muncul dapat dilihat pada gambar dibawah.

**Gambar 4. 18 wordcloud positif**




**Gambar 4. 19 wordcloud negatif**




**Gambar 4. 20 wordcloud sentimen netral**

### 4.9.3 Hasil Evaluasi Performa k-nearest neighbor

```
# --- KNN ---
k = min(40, X_train.shape[0])  # Use X_train.shape[0] to get the number of rows/samples
knn_model = KNeighborsClassifier(n_neighbors=k)
knn_model.fit(X_train, y_train)
knn_predictions = knn_model.predict(X_test)
knn_accuracy = accuracy_score(y_test, knn_predictions)
knn_report = classification_report(y_test, knn_predictions)

print("KNN Results:")
print("Accuracy:", knn_accuracy)
print("Classification Report:\n", knn_report)
```

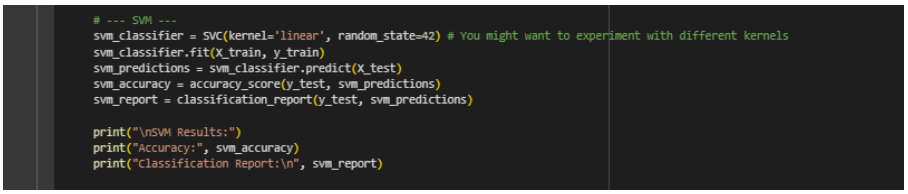
**Gambar 4. 21 code untuk menampilkan hasil evaluasi**

**Table 7 hasil performa knn**

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| 0 | 0.00 | 0.00 | 0.00 | 10 |
| 1 | 0.93 | 1.00 | 0.96 | 440 |
| 2 | 0.00 | 0.00 | 0.00 | 24 |
| Accuracy | | | 0.928 | 474 |
| Macro avg | 0.31 | 0.33 | 0.32 | 474 |
| Weighted avg | 0.86 | 0.93 | 0.89 | 474 |

Menunjukan performa untuk hasil evaluasi dari klasifikasi algoritma k-knearest neighbor berdasarkan nilai akurasi,presisi,recall dan f1-score. Berdasarkan hasil terlihat k-nearest neighbor memiliki tingkat akurasi sebesar 0,928 yang menunjukan bahwa algoritma k-knearest neighbor memiliki performa yang cukup baik dalam analisis sentimen.

### 4.9.4 Hasil Evaluasi performa support vector machine



```
# --- SVM ---
svm_classifier = SVC(kernel='linear', random_state=42) # You might want to experiment with different kernels
svm_classifier.fit(X_train, y_train)
svm_predictions = svm_classifier.predict(X_test)
svm_accuracy = accuracy_score(y_test, svm_predictions)
svm_report = classification_report(y_test, svm_predictions)

print("\nSVM Results:")
print("Accuracy:", svm_accuracy)
print("Classification Report:\n", svm_report)
```

**Gambar 4. 22 code untuk menampilkan evaluasi svm**

**Table 8 hasil evaluasi performa svm**

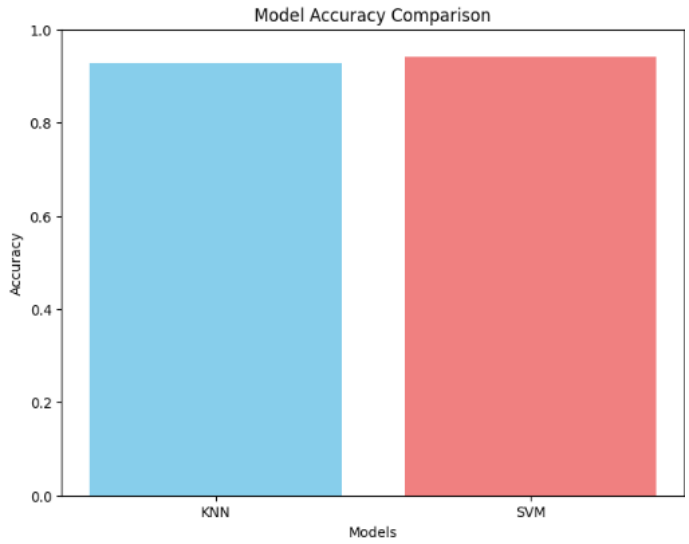
| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| 0 | 0.00 | 0.00 | 0.00 | 10 |
| 1 | 0.94 | 1.00 | 0.97 | 440 |
| 2 | 1.00 | 0.25 | 0.40 | 24 |
| Accuracy | | | 0.940 | 474 |
| Macro avg | 0.65 | 0.42 | 0.46 | 474 |
| Weighted avg | 0.92 | 0.94 | 0.92 | 474 |

Menunjukan performa untuk hasil evaluasi dari klasifikasi algoritma support vector machine berdasarkan nilai akurasi,presisi,recall dan f1-score. Berdasarkan hasil terlihat k-nearest neighbor memiliki tingkat akurasi sebesar 0,940 yang menunjukan bahwa algoritma support vector machine memiliki performa yang jauh lebih baik dari k-nearest neighbor dalam analisis sentimen.

### 4.9.5 Grafik hasil evaluasi kedua performa

```python
import matplotlib.pyplot as plt

# Assuming evaluation_results is defined as in the previous code

models = list(evaluation_results.keys())
accuracies = [evaluation_results[model]['accuracy'] for model in models]

plt.figure(figsize=(8, 6))
plt.bar(models, accuracies, color=['skyblue', 'lightcoral'])
plt.xlabel("Models")
plt.ylabel("Accuracy")
plt.title("Model Accuracy Comparison")
plt.ylim(0, 1)  # Set y-axis limit to 0-1 for accuracy
plt.show()
```

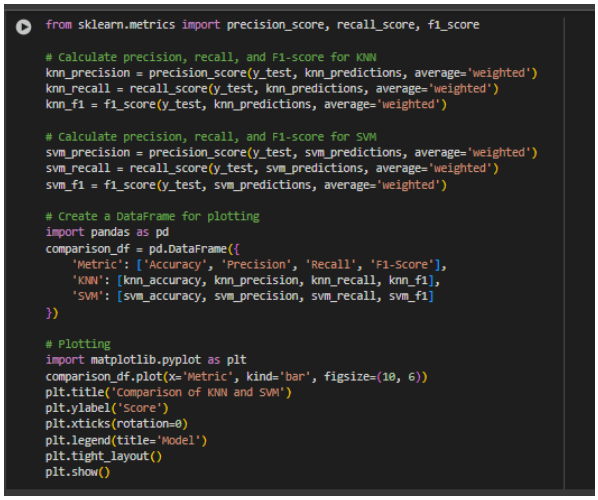
**Gambar 4. 23 code untuk menampilkan kedua performa**




**Gambar 4. 24 grafik hasil perbandingan**

Berdasarkan grafik perbandingan akurasi model, terlihat bahwa model support vector menunjukan kinerja yang lebih baik dibandingkan algoritma k-nearest neighbor dalam analisis sentimen.

### 4.9.6 Evaluasi kedua metode berdasarkan akurasi,presisi recall dan f1-score

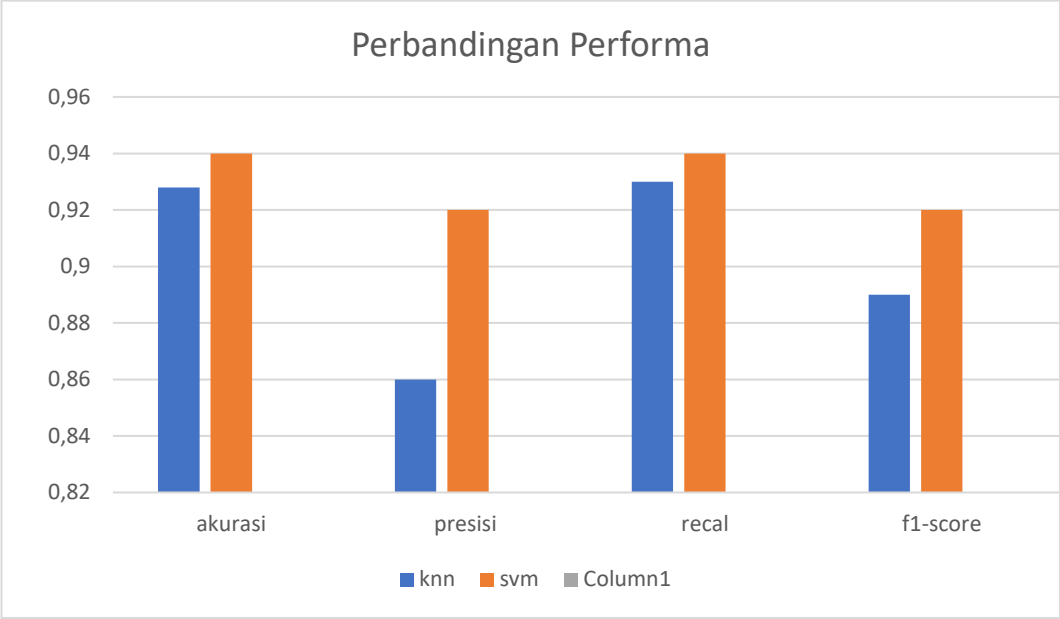


```python
from sklearn.metrics import precision_score, recall_score, f1_score

# Calculate precision, recall, and F1-score for KNN
knn_precision = precision_score(y_test, knn_predictions, average='weighted')
knn_recall = recall_score(y_test, knn_predictions, average='weighted')
knn_f1 = f1_score(y_test, knn_predictions, average='weighted')

# Calculate precision, recall, and F1-score for SVM
svm_precision = precision_score(y_test, svm_predictions, average='weighted')
svm_recall = recall_score(y_test, svm_predictions, average='weighted')
svm_f1 = f1_score(y_test, svm_predictions, average='weighted')

# Create a DataFrame for plotting
import pandas as pd
comparison_df = pd.DataFrame({
    'Metric': ['Accuracy', 'Precision', 'Recall', 'F1-Score'],
    'KNN': [knn_accuracy, knn_precision, knn_recall, knn_f1],
    'SVM': [svm_accuracy, svm_precision, svm_recall, svm_f1]
})

# Plotting
import matplotlib.pyplot as plt
comparison_df.plot(x='Metric', kind='bar', figsize=(10, 6))
plt.title('Comparison of KNN and SVM')
plt.ylabel('Score')
plt.xticks(rotation=0)
plt.legend(title='Model')
plt.tight_layout()
plt.show()
```

**Gambar 4. 25 code evaluasi model menggunakan metrik**

Code diatas Menyajikan perbandingan metrik kinerja antara model k-nearest neighbor (KNN) dan support vector machine (SVM) untuk tugas klasifikasi. Metrik yang dibandingkan meliputi akurasi presisi,recall, dan f1-score. Melalui visualisasi grafik batang, terlihat perbedaan performa antara kedua model pada masing-masing metrik. Berikut adalah hasil evaluasi kedua metode menggunakan metrik:

**Grafik perbandingan kedua perform**

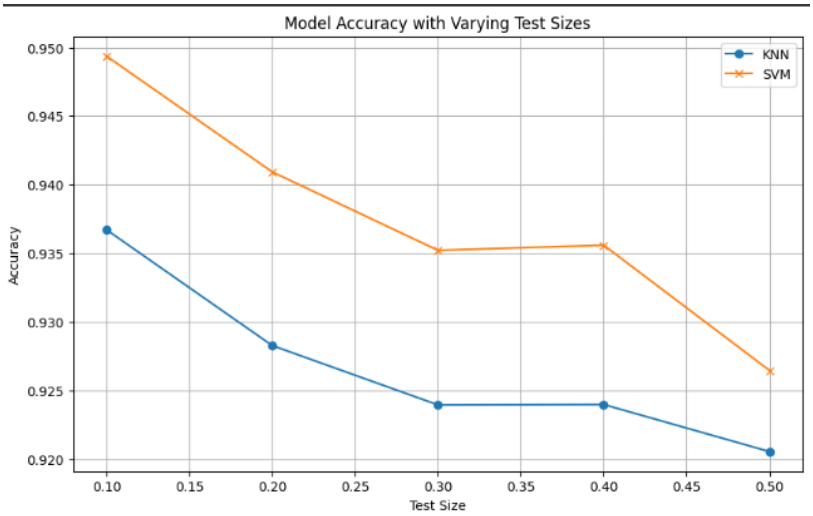



### 4.9.10 Hasil evaluasi knn dan svm dengan test size

```
# Test sizes to evaluate
test_sizes = [0.1, 0.2, 0.3, 0.4, 0.5]  # Example test sizes
evaluation_results = evaluate_models(X, y, test_sizes)

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(test_sizes, evaluation_results['KNN'], marker='o', label='KNN')
plt.plot(test_sizes, evaluation_results['SVM'], marker='x', label='SVM')
plt.xlabel('Test Size')
plt.ylabel('Accuracy')
plt.title('Model Accuracy with Varying Test Sizes')
plt.legend()
plt.grid(True)
plt.show()

# Create a DataFrame to store the results
results_df = pd.DataFrame({
    'Test Size': test_sizes,
    'KNN Accuracy': evaluation_results['KNN'],
    'SVM Accuracy': evaluation_results['SVM']
})
results_df
```

**Gambar 4. 26 code evaluasi kinerja model machine learning**

Kode ini berguna untuk mengevaluasi kinerja berbagai model berdasarkan ukuran data uji dan membandingkan secara visual.




**Gambar 4. 27 hasil test size**

Grafik ini menunjukkan akurasi model KNN (garis biru) dan SVM (garis merah) pada berbagai ukuran data uji (sumbu X).

- SVM memiliki akurasi lebih tinggi dari KNN di semua test size, tetapi akurasi keduanya menurun saat test size meningkat.
- Test size yang lebih besar mengurangi data pelatihan, sehingga akurasi menurun.
- Kesimpulan: SVM lebih unggul dibandingkan KNN dalam skenario ini.