DAFTAR PUSTAKA

Alexandridis, G., Varlamis, I., Korovesis, K., Caridakis, G., & Tsantilas, P. (2021). A survey on sentiment analysis and opinion mining in greek social media. *Information (Switzerland)*, *12*(8), 1–17. https://doi.org/10.3390/info12080331

Chaudhry, H. N., Javed, Y., Kulsoom, F., Mehmood, Z., Khan, Z. I., Shoaib, U., & Janjua, S. H. (2021). Sentiment analysis of before and after elections: Twitter data of U.S. election 2020. *Electronics (Switzerland)*, *10*(17), 1–26. https://doi.org/10.3390/electronics10172082

Chemouil, P., Hui, P., Kellerer, W., Li, Y., Stadler, R., Wen, Y., & Zhang, Y. (2019). Special Issue on Artificial Intelligence and Machine Learning for Networking and Communications. *IEEE Journal on Selected Areas in Communications*, *37*(6), 1185–1191. https://doi.org/10.1109/JSAC.2019.2909076

del Valle, E., & de la Fuente, L. (2023). Sentiment analysis methods for politics and hate speech contents in Spanish language: a systematic review. *IEEE Latin America Transactions*, *21*(3), 408–418. https://latamt.ieeer9.org/index.php/transactions/article/view/7268

Goni, M. W., Gustian, D., & Sembiring, F. (2021). Implementasi K-Means Dalam Pengelompokan Penyebaran COVID-19 di Jawa Barat. *Progresif: Jurnal Ilmiah Komputer*, *17*(2), 107. https://doi.org/10.35889/progresif.v17i2.648

Gustientiedina, G., Adiya, M. H., & Desnelita, Y. (2019). Penerapan Algoritma K-Means Untuk Clustering Data Obat-Obatan. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *5*(1), 17–24. https://doi.org/10.25077/teknosi.v5i1.2019.17-24

Haviluddin, H., Patandianan, S. J., Putra, G. M., Puspitasari, N., & Pakpahan, H. S. (2021). Implementasi Metode K-Means Untuk Pengelompokkan Rekomendasi Tugas Akhir. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *16*(1), 13.

https://doi.org/10.30872/jim.v16i1.5182

Irsyad, H., & Pribadi, M. R. (2020). Implementasi Text Mining Dalam Pengelompokan Data Tweet Pertanian Indonesia Dengan K-Means. *Kurawal - Jurnal Teknologi, Informasi Dan Industri*, *3*(2), 164–172. https://doi.org/10.33479/kurawal.v3i2.347

Kurniawan, I., & Susanto, A. (2019). Implementasi Metode K-Means dan Naïve Bayes Classifier untuk Analisis Sentimen Pemilihan Presiden (Pilpres) 2019. *Eksplora Informatika*, *9*(1), 1–10. https://doi.org/10.30864/eksplora.v9i1.237

Mahesh, B. (2020). Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR)*, *9*(1), 381–386. https://doi.org/10.21275/art20203995

Metcalf, J., Singh, R., Moss, E., Tafesse, E., & Watkins, E. A. (2023). Taking Algorithms to Courts: A Relational Approach to Algorithmic Accountability. *ACM International Conference Proceeding Series*, 1450–1462. https://doi.org/10.1145/3593013.3594092

Mihartinah, D., Anggarawati, S., & Nasution. (2023). Pengaruh Viral Marketing, Harga Dan Kepercayaan Konsumen Terhadap Keputusan Pembelian Online Melalui Media Sosial Instagram. *Student Journal of Business and Management*, *6*(1), 21–42. https://www.bps.go.id/

Nugroho, G., Murdiansyah, D. T., & Lhaksmana, K. M. (2021). Analisis Sentimen Pemilihan Presiden Amerika 2020 di Twitter Menggunakan Naïve Bayes dan Support Vector Machine. *E-Proceeding of Engineering*, *8*(5), 10106–10115.

Şekerkaya, A. (2020). Contemporary Issues in Strategic Marketing. In *Contemporary Issues in Strategic Marketing*. https://doi.org/10.26650/b/ss05.2020.002

Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm.

IEEE Access, *8*, 80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796

Tan, J. J., Firdaus, A., & Aksar, I. A. (2024). Social Media for Political Information: A Systematic Literature Review. *Jurnal Komunikasi: Malaysian Journal of Communication*, *40*(1), 77–98. https://doi.org/10.17576/JKMJC-2024-4001-05

Wirayasa, I. K. A., & Santoso, H. (2022). Analisis Employee Satisfaction Menggunakan Teknik Clustering Dan Classification Machine Learning. *Progresif: Jurnal Ilmiah Komputer*, *18*(1), 1. https://doi.org/10.35889/progresif.v18i1.766

Lampiran

1. analisa.py

- Streamlit

```python
import pandas as pd
import numpy as np
import streamlit as st
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans

# Load dataset
@st.cache
def load_data():
    return pd.read_csv('tweets_01-08-2021.csv')

# Main function
def main():
    st.title("K-Means Clustering of Tweets on Economy and Security")

    # Load data
    data = load_data()
    st.write("Dataset Loaded:")
    st.dataframe(data.head())

    # Define keywords for filtering
    economy_keywords = ['economy', 'jobs', 'growth', 'tax']
    security_keywords = ['security', 'safety', 'crime', 'law']

    # Filter relevant tweets
    data['economy_relevant'] = data['text'].apply(lambda x: any(word in x.lower() for word in economy_keywords))
    data['security_relevant'] = data['text'].apply(lambda x: any(word in x.lower() for word in security_keywords))

    economy_tweets = data[data['economy_relevant']]
    security_tweets = data[data['security_relevant']]
```

```python
# Display pie charts for economy and security tweets
st.subheader("Distribution of Relevant Tweets")
labels = ['Ec
sizes = [len(economy_tweets), len(security_tweets)]
colors = ['#ff9999','#66b3ff']

plt.figure(figsize=(8, 6))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
st.pyplot(plt)

if economy_tweets.empty and security_tweets.empty:
    st.write("No relevant tweets found.")
    return

# Preprocess data for economy tweets
if not economy_tweets.empty:
    vectorizer_economy = TfidfVectorizer(stop_words='english')
    X_economy = vectorizer_economy.fit_transform(economy_tweets['text'])

    # K-Means parameters for economy
    k_economy = st.slider("Select number of clusters for Economy (K):", 1, 10, 3)
    kmeans_economy = KMeans(n_clusters=k_economy, random_state=0)
    kmeans_economy.fit(X_economy)

    # Assign clusters
    economy_tweets['cluster'] = kmeans_economy.labels_

    # Show cluster distribution for economy
    st.subheader("Economy Cluster Distribution")
    st.write(economy_tweets['cluster'].value_counts())

    # Visualize economy clusters
    plt.figure(figsize=(10, 6))
    plt.scatter(economy_tweets['cluster'], economy_tweets['favorites'], alpha=0.5)
    plt.title("Economy Cluster Visualization based on Favorites")
    plt.xlabel("Cluster")
    plt.ylabel("Number of Favorites")
    st.pyplot(plt)

    # Display tweets in each economy cluster
    for i in range(k_economy):
        st.subheader(f"Economy Tweets in Cluster {i}")
        cluster_tweets_economy = economy_tweets[economy_tweets['cluster'] == i]
        st.write(cluster_tweets_economy[['text', 'favorites', 'date']])

# Preprocess data for security tweets
if not security_tweets.empty:
    vectorizer_security = TfidfVectorizer(stop_words='english')
    X_security = vectorizer_security.fit_transform(security_tweets['text'])

    # K-Means parameters for security
    k_security = st.slider("Select number of clusters for Security (K):", 1, 10, 3)
    kmeans_security = KMeans(n_clusters=k_security, random_state=0)
    kmeans_security.fit(X_security)

    # Assign clusters
    security_tweets['cluster'] = kmeans_security.labels_

    # Show cluster distribution for security
```

(variable) economy_tweets: DataFrame

(variable) economy_tweets: DataFrame

```
        # Assign clusters
        security_tweets['cluster'] = kmeans_security.labels_

        # Show cluster distribution for security
        st.subheader("Security Cluster Distribution")
        st.write(security_tweets['cluster'].value_counts())

        # Visualize security clusters
        plt.figure(figsize=(10, 6))
        plt.scatter(security_tweets['cluster'], security_tweets['favorites'], alpha=0.5)
        plt.title("Security Cluster Visualization based on Favorites")
        plt.xlabel("Cluster")
        plt.ylabel("Number of Favorites")
        st.pyplot(plt)

        # Display tweets in each security cluster
        for i in range(k_security):
            st.subheader(f"Security Tweets in Cluster {i}")
            cluster_tweets_security = security_tweets[security_tweets['cluster'] == i]
            st.write(cluster_tweets_security[['text', 'favorites', 'date']])

if __name__ == "__main__":
    main()
```

- Analisis.py

```
import pandas as pd
import numpy as np
import streamlit as st
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.metrics import f1_score
import nltk
from wordcloud import WordCloud
from textblob import TextBlob

# Unduh stopwords NLTK (jalankan sekali)
nltk.download('stopwords')

# Judul Aplikasi
st.title("Analisis Tweet Donald Trump")

# 1. Memuat data
data = pd.read_csv('Dataset_new_real donaldtrump_insult_tweets_2014_to_2021.csv')
tweets = data['tweet'].astype(str)

# 2. Preprocessing teks
def preprocess_text(text):
    text = text.lower()
    text = ''.join(char for char in text if char.isalpha() or char.isspace())
    return text

# Terapkan preprocessing
tweets = tweets.apply(preprocess_text)

# 3. Ekstraksi fitur menggunakan TF-IDF
from nltk.corpus import stopwords
```

```python
from nltk.corpus import stopwords
vectorizer = TfidfVectorizer(stop_words=stopwords.words('indonesian'))
X = vectorizer.fit_transform(tweets)

# 4. K-Means Clustering
num_clusters = 5
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(X)

# 5. Menambahkan hasil cl
data['cluster'] = kmeans.labels_

# 6. Menambahkan kolom tahun
data['date'] = pd.to_datetime(data['date'])
data['year'] = data['date'].dt.year

# 7. Analisis Sentimen
def get_sentiment(text):
    analysis = TextBlob(text)
    if analysis.sentiment.polarity > 0:
        return 'Positif'
    elif analysis.sentiment.polarity < 0:
        return 'Negatif'
    else:
        return 'Netral'

data['sentiment'] = data['tweet'].apply(get_sentiment)

# 8. Visualisasi

# a. Jumlah Tweet per Cluster
st.subheader('Jumlah Tweet per Cluster')
```

(variable) labels_: ndarray

```python
# a. Jumlah Tweet per Cluster
st.subheader('Jumlah Tweet per Cluster')
fig1, ax1 = plt.subplots()
data['cluster'].value_counts().plot(kind='bar', color='skyblue', ax=ax1)
ax1.set_title('Jumlah Tweet per Cluster')
ax1.set_xlabel('Cluster')
ax1.set_ylabel('Jumlah Tweet')
st.pyplot(fig1)

# b. Kata Paling Umum per Cluster
for i in range(num_clusters):
    cluster_tweets = data[data['cluster'] == i]['tweet']
    text = ' '.join(cluster_tweets)
    wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)

    st.subheader(f'Word Cloud untuk Cluster {i}')
    plt.figure(figsize=(10, 6))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    st.pyplot()

# c. Grafik Distribusi Panjang Tweet per Cluster
data['tweet_length'] = data['tweet'].apply(len)

st.subheader('Rata-rata Panjang Tweet per Cluster')
fig2, ax2 = plt.subplots()
data.groupby('cluster')['tweet_length'].mean().plot(kind='bar', color='lightcoral', ax=ax2)
ax2.set_title('Rata-rata Panjang Tweet per Cluster')
ax2.set_xlabel('Cluster')
ax2.set_ylabel('Rata-rata Panjang Tweet')
st.pyplot(fig2)
```

```python
# d. Diagram Lingkaran Jumlah Tweet per Cluster
st.subheader('Distribusi Tweet per Cluster')
fig3, ax3 = plt.subplots()
data['cluster'].value_counts().plot(kind='pie', autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors, ax=ax3)
ax3.set_title('Distribusi Tweet per Cluster')
ax3.set_ylabel('')
st.pyplot(fig3)

# e. Jumlah Tweet per Tahun
st.subheader('Jumlah Tweet per Tahun')
fig4, ax4 = plt.subplots()
data['year'].value_counts().sort_index().plot(kind='bar', color='lightgreen', ax=ax4)
ax4.set_title('Jumlah Tweet per Tahun')
ax4.set_xlabel('Tahun')
ax4.set_ylabel('Jumlah Tweet')
st.pyplot(fig4)

# f. Distribusi Sentimen
st.subheader('Distribusi Sentimen Tweet')
sentiment_counts = data['sentiment'].value_counts()

fig5, ax5 = plt.subplots()
sentiment_counts.plot(kind='bar', color=['lightblue', 'lightcoral', 'lightgreen'], ax=ax5)
ax5.set_title('Distribusi Sentimen Tweet')
ax5.set_xlabel('Sentimen')
ax5.set_ylabel('Jumlah Tweet')
ax5.set_xticklabels(sentiment_counts.index, rotation=0)
st.pyplot(fig5)

# g. Memisahkan Sentimen
st.subheader('Tweet Positif, Negatif, dan Netral')
```

```python
# Menghitung jumlah tweet berdasarkan sentimen
positive_tweets = data[data['sentiment'] == 'Positif']
negative_tweets = data[data['sentiment'] == 'Negatif']
neutral_tweets = data[data['sentiment'] == 'Netral']

st.write(f"Jumlah Tweet Positif: {len(positive_tweets)}")
st.write(f"Jumlah Tweet Negatif: {len(negative_tweets)}")
st.write(f"Jumlah Tweet Netral: {len(neutral_tweets)}")

# Menampilkan beberapa contoh tweet
st.write("Contoh Tweet Positif:")
st.write(positive_tweets['tweet'].sample(5).values)

st.write("Contoh Tweet Negatif:")
st.write(negative_tweets['tweet'].sample(5).values)

st.write("Contoh Tweet Netral:")
st.write(neutral_tweets['tweet'].sample(5).values)

# h. Menghitung F1 Score
if 'true_sentiment' in data.columns:  # Pastikan kolom ini ada
    # Mencetak F1 Score
    f1 = f1_score(data['true_sentiment'], data['sentiment'], average='weighted')
    st.write(f"F1 Score: {f1:.2f}")
else:
    st.write("Kolom 'true_sentiment' tidak ditemukan dalam dataset. F1 Score tidak dapat dihitung.")
```

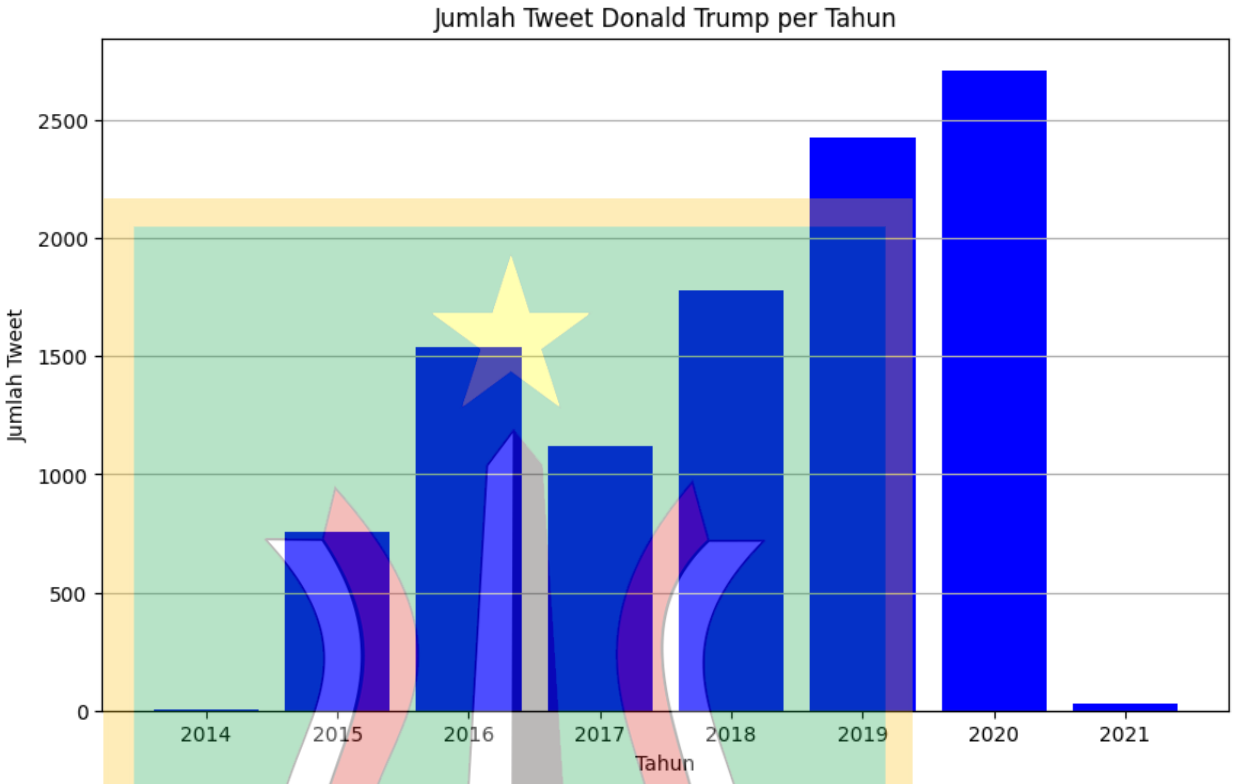
- Model

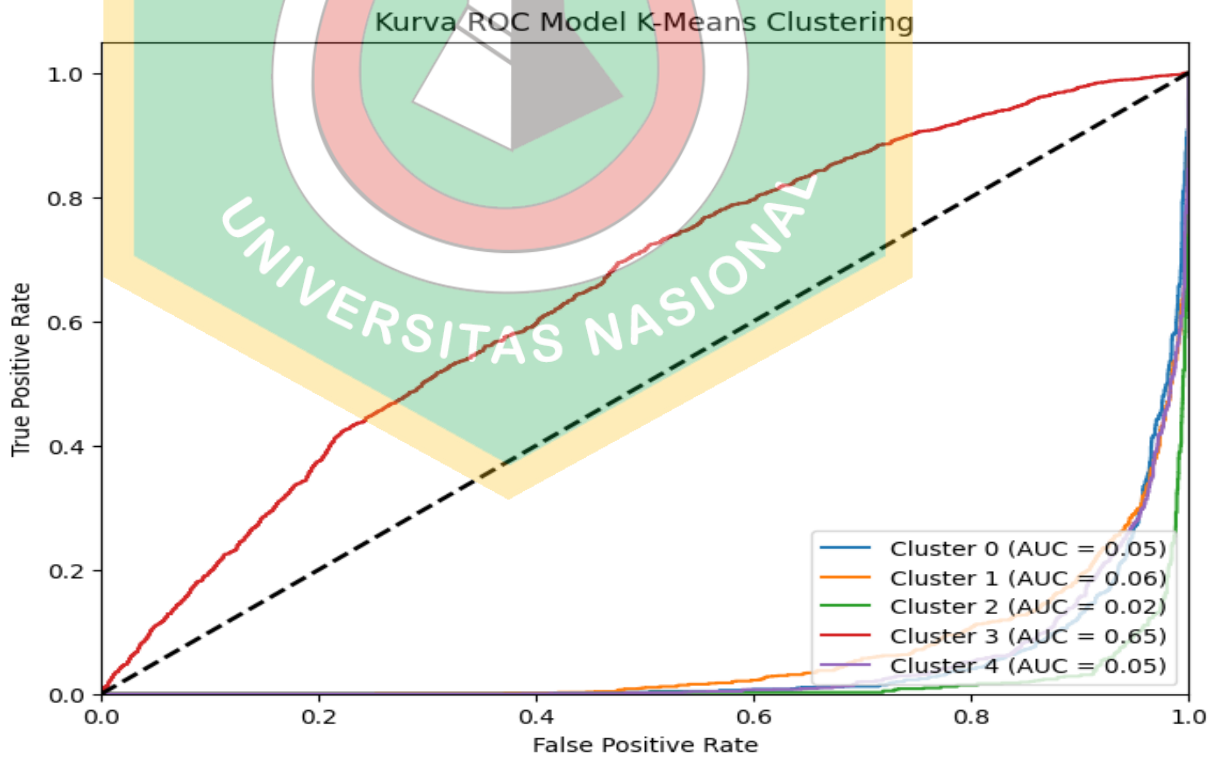
Evaluasi Model

PENGUJIAN MODEL

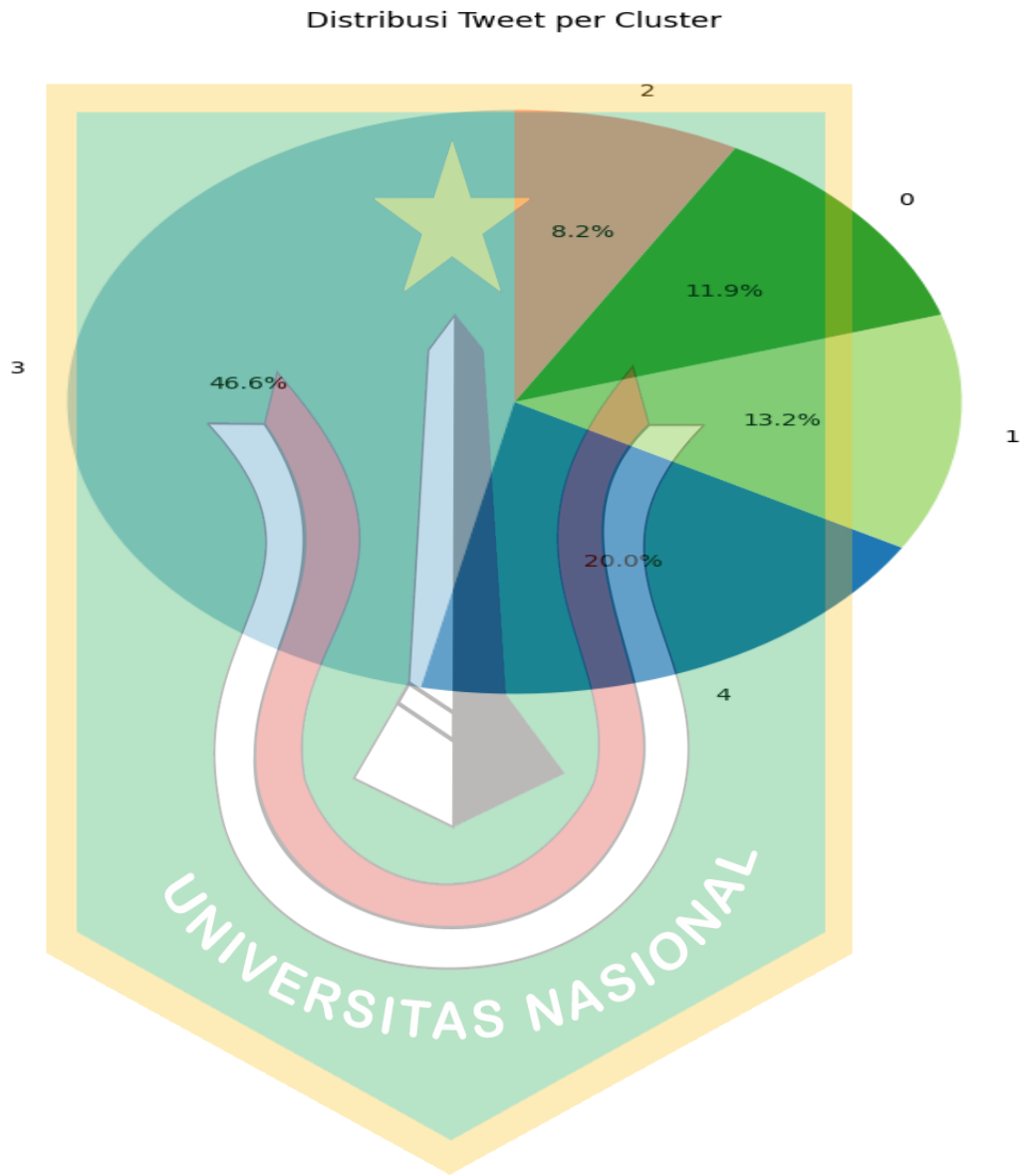
```
# Impor yang diperlukan
from sklearn.metrics import classification_report

# Hitung evaluasi model
report = classification_report(data['cluster'], kmeans.labels_)

# Tampilkan hasil
print(report)
```

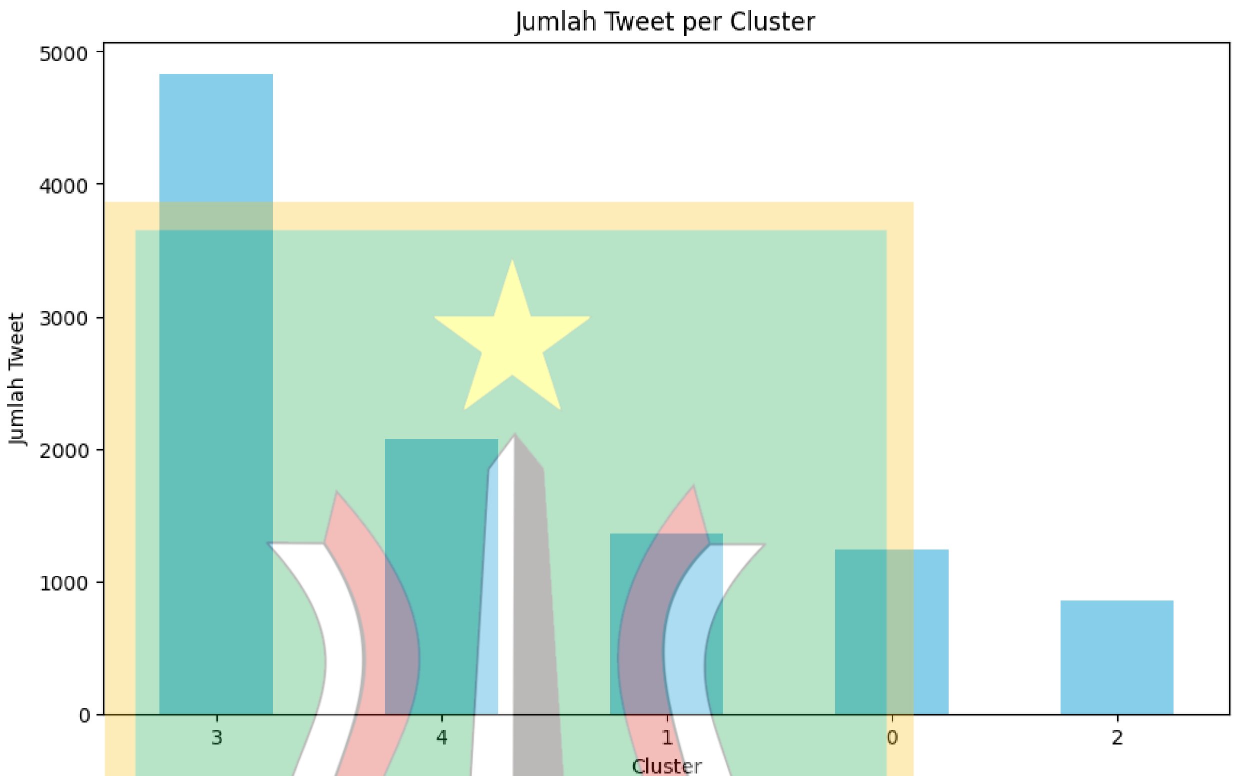
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1238
           1       1.00      1.00      1.00      1363
           2       1.00      1.00      1.00       851
           3       1.00      1.00      1.00      4832
           4       1.00      1.00      1.00      2076

    accuracy                           1.00     10360
   macro avg       1.00      1.00      1.00     10360
weighted avg       1.00      1.00      1.00     10360
```

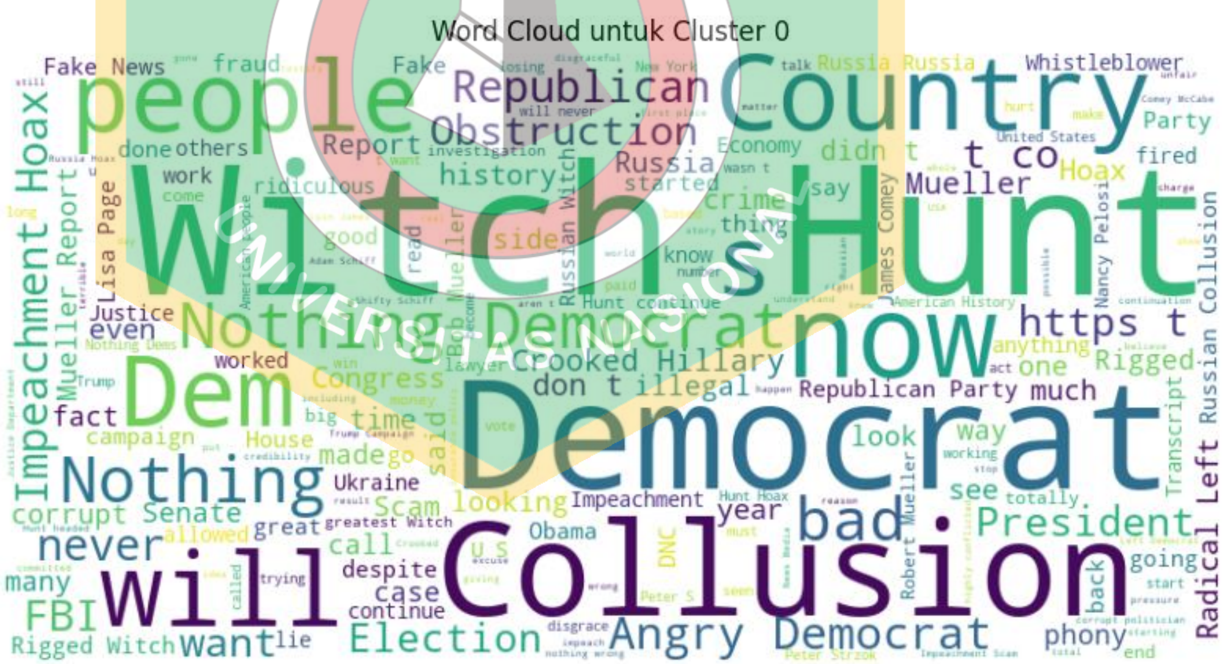




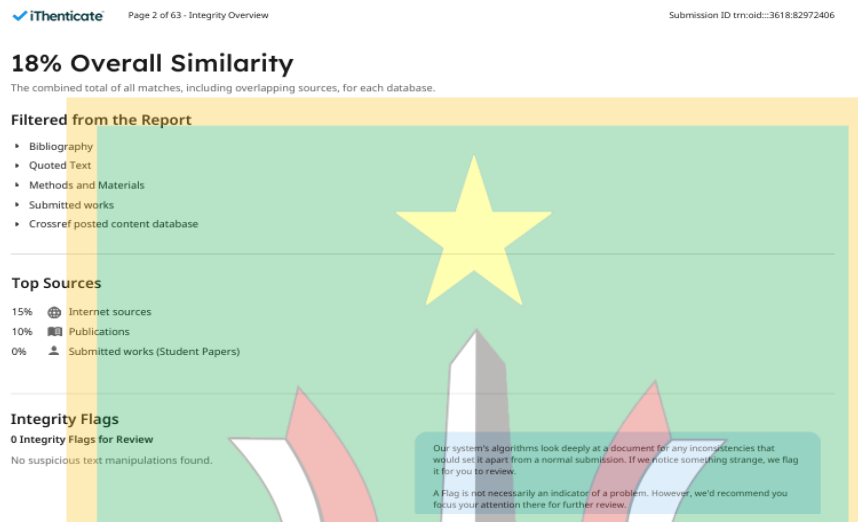
- Visualisasi Data







2. Turnitin Draf Skripsi



iThenticate Page 2 of 63 - Integrity Overview Submission ID trn:oid:::3618:82972406

18% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Methods and Materials
- Submitted works
- Crossref posted content database

Top Sources

15% Internet sources
10% Publications
0% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

3. Publish Jurnal

[JBI] Submission Acknowledgement Kotak Masuk

ojs@uajy.ac.id 17.08
kepada saya

Terjemahkan ke Indonesia

Wahyu:

Thank you for submitting the manuscript, "Publish Jurnal " to Jurnal Buana Informatika. With the online journal management system that we are using, you will be able to track its progress through the editorial process by logging in to the journal web site:

Submission URL: https://ojs.uajy.ac.id/index.php/jbi/authorDashboard/submission/11009
Username: ramadan

If you have any questions, please contact me. Thank you for considering this journal as a venue for your work.

Herlina, S.Kom., M.Eng.

Jurnal Buana Informatika

http://ojs.uajy.ac.id/index.php/jbi


UNIVERSITAS NASIONAL