

## DAFTAR PUSTAKA

- Statista. (2021). Number of Twitter users worldwide from 2019 to 2024 (in millions). Retrieved from <https://www.statista.com>.
- F. Ratnawati, "Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter," INOVTEK Polbeng - Seri Inform., vol. 3, no. 1, p. 50, 2018, doi: 10.35314/isi.v3i1.335.
- SIRINGORINGO, R. and JAMALUDDIN, 2019. Text Mining dan Klasterisasi Sentimen Pada Ulasan Produk Toko Online. *Jurnal Penelitian Teknik Informatika Universitas Prima Indonesia (UNPRI) Medan*, 2(2), pp.1–6.
- Farhan, N. M., & Setiaji, B. (2023). Indonesian Journal of Computer Science. *Indonesian Journal of Computer Science*, 12(2), 284–301. <http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135>
- Liu, B. (2020). Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Second Edition. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Second Edition*, May, 1–432. <https://doi.org/10.1017/9781108639286>
- Muhammadin, A., & Sobari, I. A. (2021). Analisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma Svm Dan Nbc. *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 2(2), 85–91. <https://doi.org/10.31294/reputasi.v2i2.785>
- Nurhidayat, R., & Dewi, K. E. (2023). Penerapan Algoritma K-Nearest Neighbor Dan Fitur Ekstraksi N-Gram Dalam Analisis Sentimen Berbasis Aspek. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, 12(1), 91–100. <https://doi.org/10.34010/komputa.v12i1.9458>
- Prasetyo, H. D., Pramiyati, T., & Isnainiyah, I. N. (2021). *Sentimen Analisis Pengguna Twitter Terhadap Kebijakan Merdeka Belajar Menggunakan Algoritma Naive Bayes*. April, 559–568.
- Putra, D. W. T., Utami, A. O., Minarni, & Swara, G. Y. (2019). Accuracy Level of Diagnosis of ENT Diseases in Expert System. *Jurnal KomtekInfo*, 6(2), 127–134. <https://doi.org/10.35134/komtekinfo.v6i2.51>
- Qadrini, L., Sepperwali, A., & Aina, A. (2021). Decision Tree Dan Adaboost Pada Klasifikasi Penerima Program Bantuan Sosial. *Jurnal Inovasi Penelitian*, 2(7), 1959–1966.
- Rahayu, S., Jaya, J. J., Mahmud, H. M., & Nugraha, F. S. (2019). Algoritma Naïve Bayes Classifier Untuk Memprediksi Gejala Autism Spectrum Disorders Pada Anak-Anak. *Seminar Nasional Rekayasa Dan Teknologi*, 27(November), 62–74.

- Ramneet, Gupta, D., & Madhukar, M. (2020). Analysis of Machine Learning Approaches for Sentiment Analysis of Twitter Data. *Journal of Computational and Theoretical Nanoscience*, 17(9), 4535–4542. <https://doi.org/10.1166/jctn.2020.9300>
- Septianingrum, F., & Irawan, A. S. Y. (2021). Metode Seleksi Fitur Untuk Klasifikasi Sentimen Menggunakan Algoritma Naive Bayes: Sebuah Literature Review. *Jurnal Media Informatika Budidarma*, 5(3), 799. <https://doi.org/10.30865/mib.v5i3.2983>
- Supian, A., Tri Revaldo, B., Marhadi, N., Efrizoni, L., & Rahmadden, R. (2024). Perbandingan Kinerja Naïve Bayes Dan Svm Pada Analisis Sentimen Twitter Ibukota Nusantara. *Jurnal Ilmiah Informatika*, 12(01), 15–21. <https://doi.org/10.33884/jif.v12i01.8721>
- Suryati, E., Styawati, & Aldino, A. A. (2023). Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Dan Sistem Informasi*, 4(1), 96–106. <https://doi.org/10.33365/jtsi.v4i1.2445>
- Troussas, C., Virvou, M., Espinosa, K. J., Llaguno, K., & Caro, J. (2013). Sentiment analysis of Facebook statuses using Naive Bayes Classifier for language learning. *IISA 2013 - 4th International Conference on Information, Intelligence, Systems and Applications*, 198–205. <https://doi.org/10.1109/IISA.2013.6623713>
- Widowati, T. T., & Sadikin, M. (2021). Analisis Sentimen Twitter terhadap Tokoh Publik dengan Algoritma Naive Bayes dan Support Vector Machine. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 11(2), 626–636. <https://doi.org/10.24176/simet.v11i2.4568>
- L. Qadrini, A. Sepperwali, and A. Aina, “Decision Tree Dan Adaboost Pada Klasifikasi Penerima Program Bantuan Sosial,” *J. Inov. Penelit.*, vol. 2, no. 7, pp. 1959–1966, 2021.
- Ohana B, Tierney B. Sentiment classification of reviews using SentiWordNet. In: 9th International Conference on Information Technology and Telecommunication: Ubiquitous and Green Computing [Internet]. 2009. p. 3–10. Available from: <http://www.ittconference.ie/openconf/openconf.php>

# LAMPIRAN

## 1. Streamlit

```
# Fungsi utama Streamlit
@st.cache_resource
def dataset():
    df = pd.read_csv('labeled_NadiemMakarim.csv')
    st.dataframe(df, use_container_width=True)
    st.write(f"Jumlah data: {len(df)}")
    if 'text' in df.columns:
        # Membersihkan teks di dalam kolom 'text'
        df['cleaned_text'] = df['text'].apply(lambda text: re.sub(r'\d+', '', text)) # Hapus
        angka
        df['cleaned_text'] = df['cleaned_text'].apply(lambda text: re.sub(r'^\w\s', '',
        text)) # Hapus tanda baca
        df['cleaned_text'] = df['cleaned_text'].apply(lambda text: text.lower()) # Case
        folding
        df['cleaned_text'] = df['cleaned_text'].apply(lambda text: ' '.join([word for word
        in word_tokenize(text) if word not in stopwords.words('indonesian')])) # Hapus
        stopwords

        # Tampilkan hasil pembersihan teks
        st.subheader("Hasil Pembersihan Teks")
        st.dataframe(df[['text', 'cleaned_text']])

        # Tampilkan jumlah data yang bersih (jumlah data yang memiliki cleaned_text)
        clean_data_count = df['cleaned_text'].str.strip().ne("").sum() # Menghitung
        jumlah teks yang tidak kosong setelah dibersihkan
        st.write(f"Jumlah data yang telah dibersihkan: {clean_data_count}")

        # Vektorisasi teks menggunakan TF-IDF
        vectorizer = TfidfVectorizer()
        X = vectorizer.fit_transform(df['cleaned_text']) # Data teks sebagai fitur
        y = df['sentiment'] # Label sebagai target

        # Menampilkan dimensi hasil vektorisasi
        st.write("Dimensi fitur (TF-IDF):", X.shape)

        # Tampilkan beberapa fitur pertama dari vektorisasi
        st.subheader("Beberapa fitur pertama dari TF-IDF:")
        features = vectorizer.get_feature_names_out()
        st.write(features)

        # Bagi data menjadi training dan testing
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
        random_state=42)

        # Menyimpan data latih dan uji ke dalam session state
```

```

st.session_state.X_train = X_train
st.session_state.X_test = X_test
st.session_state.y_train = y_train
st.session_state.y_test = y_test

# Tampilkan jumlah data pada masing-masing set
st.write(f"Jumlah data latih: {X_train.shape[0]}")
st.write(f"Jumlah data uji: {X_test.shape[0]}")

# Menampilkan data latih dan data uji
st.subheader("Data Latih (X_train) dan Label Latih (y_train):")
st.write(X_train.toarray()[:5]) # Menampilkan 5 data pertama dari X_train
st.write(y_train.head()) # Menampilkan 5 label pertama dari y_train

st.subheader("Data Uji (X_test) dan Label Uji (y_test):")
st.write(X_test.toarray()[:5]) # Menampilkan 5 data pertama dari X_test
st.write(y_test.head()) # Menampilkan 5 label pertama dari y_test

def hasil():
    #Sesion
    X_train = st.session_state.X_train
    X_test = st.session_state.X_test
    y_train = st.session_state.y_train
    y_test = st.session_state.y_test

    # Melatih model Naive Bayes
    nb_model = MultinomialNB()
    nb_model.fit(X_train, y_train)
    nb_predictions = nb_model.predict(X_test)

    # Evaluasi model
    st.subheader("Evaluasi Model Naive Bayes")
    st.text("Classification Report:")
    report = classification_report(y_test, nb_predictions, output_dict=True)
    report_df = pd.DataFrame(report).transpose()
    report_html = report_df.style.set_table_attributes("class='classification-report'").format(precision=2).to_html()
    st.markdown(report_html, unsafe_allow_html=True)
    st.write("Accuracy:", accuracy_score(y_test, nb_predictions))

    #Sesion
    # Melatih model SVM dengan kernel linear
    svm_model = SVC(kernel='linear')
    svm_model.fit(X_train, y_train)

    # Prediksi data uji
    svm_predictions = svm_model.predict(X_test)

    # Evaluasi model
    st.subheader("Evaluasi Model SVM")

```

```

# Mengubah classification_report menjadi DataFrame
report = classification_report(y_test, svm_predictions, output_dict=True)
report_df = pd.DataFrame(report).transpose()

# Menampilkan classification report dalam bentuk tabel HTML
report_html = report_df.style.set_table_attributes("class='classification-
report']").format(precision=2).to_html()
st.write("Classification Report:")
st.markdown(report_html, unsafe_allow_html=True)
st.write("Accuracy:", accuracy_score(y_test, svm_predictions))

# Data hasil evaluasi
metrics = ['Precision', 'Recall', 'F1-Score', 'Accuracy']
nb_scores = [classification_report(y_test, nb_predictions,
output_dict=True)['weighted avg'][metric.lower()] for metric in metrics[:-1]]
nb_scores.append(accuracy_score(y_test, nb_predictions))

svm_scores = [classification_report(y_test, svm_predictions,
output_dict=True)['weighted avg'][metric.lower()] for metric in metrics[:-1]]
svm_scores.append(accuracy_score(y_test, svm_predictions))

# Grafik perbandingan
x = np.arange(len(metrics))
width = 0.35

# Desain grafik menggunakan Streamlit
st.subheader("Perbandingan Hasil Evaluasi Model")
st.write("Grafik berikut ini menunjukkan perbandingan hasil evaluasi antara model
Naive Bayes dan SVM.")

fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(x - width/2, nb_scores, width, label='Naive Bayes', color='skyblue')
ax.bar(x + width/2, svm_scores, width, label='SVM', color='lightgreen')

ax.set_ylabel('Score')
ax.set_title('Perbandingan Hasil Evaluasi Model Naive Bayes dan SVM')
ax.set_xticks(x)
ax.set_xticklabels(metrics)
ax.legend()

# Menampilkan grafik di Streamlit
st.pyplot(fig)

# Membaca dataset
df = pd.read_csv('labeled_NadiemMakarim.csv')

# Menampilkan jumlah setiap jenis sentimen
sentiment_counts = df['sentiment'].value_counts()

```

```

# Visualisasi distribusi sentimen
st.subheader("Distribusi Sentimen pada Tweet Nadiem Makarim")
plt.figure(figsize=(10, 6))
sns.barplot(x=sentiment_counts.index, y=sentiment_counts.values, palette='viridis')
plt.title('Distribusi Sentimen pada Tweet Nadiem Makarim')
plt.xlabel('Sentimen')
plt.ylabel('Jumlah Tweet')
plt.xticks(rotation=45)

# Menampilkan grafik di Streamlit
st.pyplot(plt)

# Grafik favorite counts berdasarkan sentiment
st.subheader("Total Favorite Counts Berdasarkan Sentimen")
plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='sentiment', y='favorite_count', palette='coolwarm',
estimator=sum)
plt.title("Total Favorite Counts by Sentiment")
plt.xlabel('Sentiment')
plt.ylabel('Total Favorite Count')
plt.xticks(rotation=45)

# Menampilkan grafik di Streamlit
st.pyplot(plt)

# Menyaring data berdasarkan sentimen positif, negatif, dan netral
positive_tweets = df[df['sentiment'] == 'positif']
negative_tweets = df[df['sentiment'] == 'negatif']
neutral_tweets = df[df['sentiment'] == 'netral']

# Menambahkan kolom untuk menandai tweet yang mengandung kata kunci
positif, negatif, dan netral
df['contains_positive'] = df['text'].apply(lambda x: any(keyword in x.lower() for
keyword in positive_tweets))
df['contains_negative'] = df['text'].apply(lambda x: any(keyword in x.lower() for
keyword in negative_tweets))
df['contains_neutral'] = df['text'].apply(lambda x: any(keyword in x.lower() for
keyword in neutral_tweets))

# Visualisasi distribusi frekuensi kata kunci berdasarkan sentimen
st.subheader("Distribusi Sentimen Berdasarkan Kata Kunci Positif")

# Plotting
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='sentiment', hue='contains_positive', palette='Blues',
hue_order=[True, False])
plt.title('Distribusi Sentimen Berdasarkan Kata Kunci Positif')
plt.xlabel('Sentimen')
plt.ylabel('Jumlah Tweet')
plt.legend(title='Mengandung Kata Kunci Positif', labels=['Tidak', 'Ya'])

```

```

# Menampilkan grafik di Streamlit
st.pyplot(plt)

# Mengonversi kolom 'created_at' ke tipe data datetime
df['created_at'] = pd.to_datetime(df['created_at'])

# Menetapkan 'created_at' sebagai index
df.set_index('created_at', inplace=True)

# Membuat visualisasi tren favorite count dari waktu ke waktu
st.title("Trend of Favorite Counts Over Time")
st.write("Grafik ini menunjukkan total jumlah favorite berdasarkan waktu tweet.")

# Menggunakan resample untuk menghitung jumlah favorite per hari
daily_favorites = df.resample('D')['favorite_count'].sum()

# Plotting menggunakan Matplotlib
plt.figure(figsize=(14, 7))
daily_favorites.plot()

# Menambahkan judul dan label
plt.title("Trend of Favorite Counts Over Time")
plt.xlabel('Date')
plt.ylabel('Total Favorite Count')
plt.grid(True)

# Menampilkan grafik di Streamlit
st.pyplot(plt)

return hasil

```

## 2. Hasil Turnitin

UNIVERSITAS NASIONAL



ORIGINALITY REPORT

**20%** SIMILARITY INDEX  
**16%** INTERNET SOURCES  
**13%** PUBLICATIONS  
**6%** STUDENT PAPERS

PRIMARY SOURCES

|   |                                                       |     |
|---|-------------------------------------------------------|-----|
| 1 | ijcs.stmikindonesia.ac.id                             | 1%  |
| 2 | kc.umn.ac.id                                          | 1%  |
| 3 | digilib.unila.ac.id                                   | <1% |
| 4 | garuda.kemdikbud.go.id                                | <1% |
| 5 | Submitted to Universitas Pertamina                    | <1% |
| 6 | Submitted to University of North Carolina, Greensboro | <1% |
| 7 | Submitted to Universitas Maritim Raja Ali Haji        | <1% |

### 3. Publikasi Jurnal

UJUMATICS: Jurnal Teknik Informatika

Submission Library View Metadata

The TWITTER SENTIMENT ANALYSIS ON NADIEM MAKARIM USING SVM AND NAÏVE BAYES ALGORITHMS

Mahesa Dias Seputra

Submission Review Copyediting Production

Submission Files

17726-1 mahesadias29@gmail.com, Author, TWITTER SENTIMENT ANALYSIS ON NADIEM MAKARIM USING SVM AND NAÏVE BAYES ALGORITHMS.doc Article Text

Download All Files

Pre-Review Discussions

No Items

Platform & workflow by OJS / PKP

UNIVERSITAS NASIONAL