# DAFTAR PUSTAKA


Ahmadi, M. I., Gustian, D., & Sembiring, F. (2021). Analisis Sentiment Masyarakat terhadap Kasus Covid-19 pada Media Sosial Youtube dengan Metode Naive bayes. In *Jurnal Sains Komputer & Informatika (J-SAKTI* (Vol. 5, Issue 2).

Alfarizi, S., & Fitriani, E. (2023). Analisis Sentimen Kendaraan Listrik Menggunakan Algoritma Naive Bayes dengan Seleksi Fitur Information Gain dan Particle Swarm Optimization. *Indonesian Journal on Software Engineering (IJSE)*, *9*(1), 19–27. http://ejournal.bsi.ac.id/ejurnal/index.php/ijse

Ananda, N. (2023). *Klasifikasi Sentimen Tweet Masyarakat Terhadap Kendaraan Listrik Menggunakan Support Vector Machine*. https://doi.org/10.32493/informatika.v8i4.36754

Darmawan Sidik, A., Ansawarman, A., Kunci, K., Kendaraan Bermotor, J., Regresi, M., & Jalan, F. (2022). Prediksi Jumlah Kendaraan Bermotor Menggunakan Machine Learning. *Formosa Journal of Multidisciplinary Research (FJMR)*, *1*(3), 559–568. https://doi.org/10.55927

Elektronika, J., & Komputer, D. (2023). *Perbandingan Naïve Bayes dan KNN Dalam Klasifikasi Tweet BBM Subsidi*. *16*(1), 35–44. https://journal.stekom.ac.id/index.php/elkom■page35

Fauzi Fayyad, M., Takratama Savra, D., Kurniawan, V., & Hilmi Estanto, B. (2023). *Sentiment Analysis of Towards Electric Cars using Naive Bayes Classifier and Support Vector Machine Algorithm*. *1*(1), 1–9. https://journal.irpi.or.id/index.php/predatecs/article/view/814

Fazal, R., Andraini, L., & Komputer, T. (n.d.). Membandingkan Support Vector Machines Dan Naive Bayes Pada Analisis Sentimen Data Twitter. In *Portaldata.org* (Vol. 2, Issue 10).

Fitriyana, V., Hakim, L., Candra Rini Novitasari, D., Hanif Asyhar, A., Studi Matematika, P., Sains Dan Teknologi, F., Sunan Ampel Surabaya, U., & Timur, J. (2023). Analisis Sentimen Ulasan Aplikasi Jamsostek Mobile Menggunakan Metode Support Vector Machine. In *Jurnal Buana Informatika* (Vol. 14, Issue 1).

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems)*.

Hanyokro Kusuma, G., Permana, I., Nur Salisah, F., Afdal, M., Jazman, M., Marsal, A., Studi Sistem Informasi, P., & Sains dan Teknologi, F. (n.d.). *Pendekatan Machine Learning: Analisis Sentimen Masyarakat Terhadap Kendaraan Listrik Pada Sosial Media X*. *9*(2), 65–76.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Nuriy Alin, F., Hafid Totohendarto, M., & Rafi Muttaqin, M. (n.d.). Analisis Sentimen Terhadap Kendaraan Listrik Pada Platform Twitter Menggunakan Metode Naive

Bayes. *Informatics for Educators And Professionals : Journal of Informatics*, *8*(2), 96–107.

Pamungkas, F. S., & Kharisudin, I. (2021). *Analisis Sentimen dengan SVM*. *4*, 628–634. https://journal.unnes.ac.id/sju/index.php/prisma/

Poedjimartojo, N. A., Pramesti, D., & Fa'rifah, R. Y. (2023). Sentiment analysis on public opinion of electric vehicles usage in Indonesia using support vector machine algorithms. *Teknika: Jurnal Sains Dan Teknologi*, *19*(2), 152. https://doi.org/10.36055/tjst.v19i2.21967

Prabowo, W. A., & Wiguna, C. (2021). Sistem Informasi UMKM Bengkel Berbasis Web Menggunakan Metode SCRUM. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *5*(1), 149. https://doi.org/10.30865/mib.v5i1.2604

Pravina, A. M., Cholissodin, I., & Adikara, P. P. (2019). *Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM)* (Vol. 3, Issue 3). http://j-ptiik.ub.ac.id

Putu Gita Naraswati, N., Cindy Rosmilda, D., Desinta, D., Khairi, F., Damaiyanti, R., & Nooraeni, R. (n.d.). *SISTEMASI: Jurnal Sistem Informasi Analisis Sentimen Publik dari Twitter Tentang Kebijakan Penanganan Covid-19 di Indonesia dengan Naive Bayes Classification*. http://sistemasi.ftik.unisi.ac.id

Rahat, A. M., Kahir, A., & Masum, A. K. M. (2020). Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset. *Proceedings of the 2019 8th International Conference on System Modeling and Advancement in Research Trends, SMART 2019*, 266–270. https://doi.org/10.1109/SMART46866.2019.9117512

*SALINAN PRESIDEN BLIK INDONESIA PERATURAN PRESIDEN REPUBLIK INDONESIA Menimbang Mengingat*. (n.d.).

Septiani Hudaya, C., Fakhrurroja, H., & Alamsyah, A. (2019). Andry Alamsyah 3) 664 Cindy Septiani Hudaya 1). *Cindy Septiani Hudaya*, *1*(6), 664–673.

Siregar, A. M. (2023). Analisis Sentimen Pindah Ibu Kota Negara (IKN) Baru pada Twitter Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM). *Faktor Exacta*, *16*(3). https://doi.org/10.30998/faktorexacta.v16i3.16703

Wicaksono, A. S. (2023). TWITTER SENTIMENT ANALYSIS OF ELECTRIC VEHICLE SUBSIDY POLICY USING NAÏVE BAYES ALGORITHM. In *Journal of Statistical Methods and Data Science* (Vol. 01, Issue 1). https://jurnalmipa.unri.ac.id/jsmds

Yoliadi, D, N. (2023). Data mining Dalam Analisis Tingkat Penjualan Barang Elektronik Menggunakan Algoritma K-means. Insearch (Information System Research) Journal, 3(1), 43-53.

# LAMPIRAN

1. Pengujian Kalimat Baru

| No | Kalimat Baru | NB | SVM |
|---|---|---|---|
| 1 | The government's electric vehicle policy is a great step toward reducing carbon emissions | Positif | Positif |
| 2 | The transition to electric vehicles depends on infrastructure and consumer acceptance | Positif | Netral |
| 3 | Electric vehicles have high maintenance costs, making them less practical for average users | Negatif | Negatif |
| 4 | The electric vehicle policy supports innovation and the development of sustainable transportation | Positif | Positif |
| 5 | Electric vehicle policies vary across different countries and regions | Netral | Netral |
| 6 | The transition to electric vehicles depends on infrastructure development and consumer demand | Positif | Netral |
| 7 | Expanding charging stations will make electric vehicles more accessible and convenient | Positif | Positif |
| 8 | Limited charging stations make it difficult for people to rely on EVs for daily use | Netral | Negatif |
| 9 | Some people support EV policies, while others remain skeptical about their effectiveness | Positif | Netral |
| 10 | Poor people government subsidizes electric vehicles | Negatif | Negatif |
| 11 | Incentives for EV buyers encourage more people to switch to environmentally friendly transportation | Positif | Positif |
| 12 | The cost of installing a home charging station is too expensive for most people | Negatif | Negatif |
| 13 | There are ongoing discussions about the benefits and challenges of EV adoption | Netral | Netral |
| 14 | Electric vehicle incentives only benefit the wealthy, while lower income groups are left behind | Negatif | Negatif |
| 15 | Many people are hesitant to switch to electric vehicle due to unclear long term policies | Positif | Negatif |

2. Source Code Python Pada Google Colab

```
import pandas as pd
df = pd.read_csv("data sentiment1.csv", sep=";")
df.head()


#Mengahapus data duplicate
df.drop_duplicates(subset="full_text", keep = 'first', inplace =True)
df.info()


#Cleaning Data
import re
import string
import nltk

def remove_hashtags(text):
    return re.sub(r"(#\w+)", "", text).strip()

def remove_URL(tweet):
  if tweet is not None and isinstance(tweet, str):
    url = re.compile(r'https?://\S+|www\.\S+')
    return url.sub(r'', tweet)
  else:
    return tweet

def remove_html(tweet):
  if tweet is not None and isinstance(tweet, str):
    html = re.compile(r'<.*?>')
    return html.sub(r'', tweet)
  else:
    return tweet

def remove_emoji(tweet):
  if tweet is not None and isinstance(tweet, str):
     emoji_pattern = re.compile("["
     u"\U0001F600-\U0001F64F"
      u"\U0001F300-\U0001F5FF"
      u"\U0001F680-\U0001F6FF"
      u"\U0001F700-\U0001F77F"
      u"\U0001F780-\U0001F7FF"
      u"\U0001F800-\U0001F8FF"
      u"\U0001F900-\U0001F9FF"
      u"\U0001FA00-\U0001FA6F"
      u"\U0001FA70-\U0001FAFF"
      u"\U0001F004-\U0001F0CF"
      u"\U0001F1E0-\U0001F1FF"
     "]", flags=re.UNICODE)

     return emoji_pattern.sub(r'', tweet)
```

```
  else:
    return tweet

def remove_symbols(tweet):
  if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'[^a-zA-Z0-9\s]', '', tweet)
        return tweet

def remove_numbers(tweet):
  if tweet is not None and isinstance(tweet, str):
    tweet = re.sub(r'\d', '', tweet)
    return tweet

def remove_username(text):
  if text is not None and isinstance(text, str):
    import re
    return re.sub(r'@[^\s]+', '', text)

def clean_text(text):
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)
    text = re.sub(r'\d+', '', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = text.strip()
    text = re.sub(r'[\U00010000-\U0010FFFF]', '', text, flags=re.UNICODE)
    text = re.sub(r'[\x00-\x1f\x7f-\x9f\u2000-\u2fff\u3000-\u303f]', '', text)
    text = re.sub(r'[^\x00-\x7F]+', '', text)
    text = re.sub(r'[\x00-\x1F\x7F-\x9F]', '', text)
    text = re.sub(r'[^\w\s]', '', text)

df['cleaning'] = df['full_text'].apply(lambda x: remove_URL(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_username(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_hashtags(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_html(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_emoji(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_symbols(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_numbers(x))
df['cleaning'] = df['cleaning'].apply(lambda x: clean_text(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_hashtags(x))


#Case Folding
def case_folding(text):
  if isinstance(text, str):
    lowercase_text = text.lower()
    return lowercase_text
  else:
    return text

df['case_folding'] = df['cleaning'].apply(case_folding)
```

```
#Normalisasi
!pip install openpyxl
import pandas as pd

kamus_data = pd.read_excel('/content/slang_dict.xlsx', engine='openpyxl')
kamus_tidak_baku = dict(zip(kamus_data['kata tidak baku'], kamus_data['kata baku']))
df[['normalisasi', 'data_baku', 'kata_tidak_baku', 'kata_tidak_baku_hash']] = pd.DataFrame([replace_taboo_words(x, kamus_tidak_baku) for x in df['case_folding'] ], index=df.index)


#Tokenize
def tokenize(text):
  tokens = text.split()
  return tokens
df['tokenize'] = df['normalisasi'].apply(tokenize)


#Stopwords
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = set(stopwords.words('indonesian'))

def remove_stopwords(text):
  return [word for word in text if word not in stop_words]
df['stopword removal'] = df['tokenize'].apply(lambda x: remove_stopwords(x))


#Stemming
!pip install Sastrawi
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer

def stem_text(words):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    if isinstance(words, list):
        return ' '.join([stemmer.stem(word) for word in words])
    else:
        return ' '.join([stemmer.stem(word) for word in words.split()])

df['stemming_data'] = df['stopword removal'].apply(stem_text)


#Translate
!pip install deep-translator
```

```
from deep_translator import GoogleTranslator

def convert_eng(text):
  translator = GoogleTranslator()
  translatation = translator.translate(text, lang_tgt='en', lang_src="id")
  return translatation
df['translate'] = df['stemming_data'].apply(convert_eng)


#Labeling
!pip install textblob
!pip install wordcloud
from textblob import TextBlob

label_df = list(df['translate'])
polaritas = 0

status = []
total_positif = total_negatif = total_netral = total = 0

for i, data in enumerate(label_df):
    blob = TextBlob(str(data))
    sentiment = blob.sentiment.polarity
    polaritas += sentiment

    if sentiment > 0.0:
        total_positif += 1
        status.append("positive")
    elif sentiment < 0.0:
        total_negatif += 1
        status.append("negative")
    else:
        total_netral += 1
        status.append("neutral")

    total += 1


#TF-IDF & Split Data
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
x_train, x_test, y_train, y_test = train_test_split(df['translate'], df['sentimen'],
test_size=0.2, random_state=42)

vectorizer = TfidfVectorizer()
x_train_vectorized = vectorizer.fit_transform(x_train)
x_test_vectorized = vectorizer.transform(x_test)

# Klasifikasi NB
```

```
from sklearn.naive_bayes import ComplementNB
model = ComplementNB()
model.fit(x_train_vectorized, y_train)
predictions = model.predict(x_test_vectorized)
accuracy = accuracy_score(y_test, predictions)
import seaborn as sns
import matplotlib.pyplot as plt
conf_matrix = confusion_matrix(y_test, predictions)

# Klasifikasi SVM
from sklearn.svm import SVC
feature_names = vectorizer.get_feature_names_out()
svm_model = SVC(kernel='linear', C=10, random_state=42)
svm_model.fit(x_train_vectorized, y_train)
y_pred = svm_model.predict(x_test_vectorized)

accuracy =accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

3. Source Code Python untuk Streamlit

```
import streamlit as st
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import ComplementNB
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import numpy as np

st.title("📊 Dashboard Visualisasi Analisis Sentimen Kebijakan Kendaraan
Listrik")

uploaded_file = st.file_uploader("Unggah file CSV", type=["csv"])

if uploaded_file is not None:
        df = pd.read_csv(uploaded_file, delimiter=';')
        df.dropna(subset=['translate', 'sentimen'], inplace=True)
        st.subheader("Dataset")
        st.write("Berikut tampilan dataset terkait kebijakan kendaraan listrik")
        st.dataframe(df, height=300)

if 'translate' in df.columns and 'sentimen' in df.columns:
        st.subheader("Distribusi Kelas Sentimen")
```

```
        st.write("Berikut merupakan visualisasi distribusi kelas sentimen yang terbagi menjadi 3 kelas")
        fig, ax = plt.subplots()
        sns.countplot(x=df['sentimen'], ax=ax)
        st.pyplot(fig)

# Split data menjadi training dan testing set
X_train, X_test, y_train, y_test = train_test_split(df['translate'], df['sentimen'], test_size=0.2, random_state=42)


# Visualisasi jumlah data latih dan uji menggunakan bar chart
st.subheader("Pembagian Data Latih dan Uji")
st.write("Berikut merupakan visualisasi pembagian data latih 80 dan data uji 20")
sizes = [len(X_train), len(X_test)]
labels = ['Data Latih', 'Data Uji']


fig, ax = plt.subplots()
ax.bar(labels, sizes, color=['#66b3ff', '#99ff99'])
ax.set_ylabel('Jumlah Data')
ax.set_title('Jumlah Data Latih dan Uji')
st.pyplot(fig)


vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)


# Model Naïve Bayes Multinomial
nb_model = ComplementNB()
nb_model.fit(X_train_tfidf, y_train)
y_pred_nb = nb_model.predict(X_test_tfidf)
acc_nb = accuracy_score(y_test, y_pred_nb)
cm_nb = confusion_matrix(y_test, y_pred_nb)


# Model SVM
svm_model = SVC(kernel='linear', C=10)
svm_model.fit(X_train_tfidf, y_train)
y_pred_svm = svm_model.predict(X_test_tfidf)
acc_svm = accuracy_score(y_test, y_pred_svm)
cm_svm = confusion_matrix(y_test, y_pred_svm)


st.subheader("Akurasi Model")
st.write(f"🎯 Naïve Bayes: {acc_nb:.2f}")
st.write(f"🎯 SVM: {acc_svm:.2f}")


# Klasifikasi Report
```

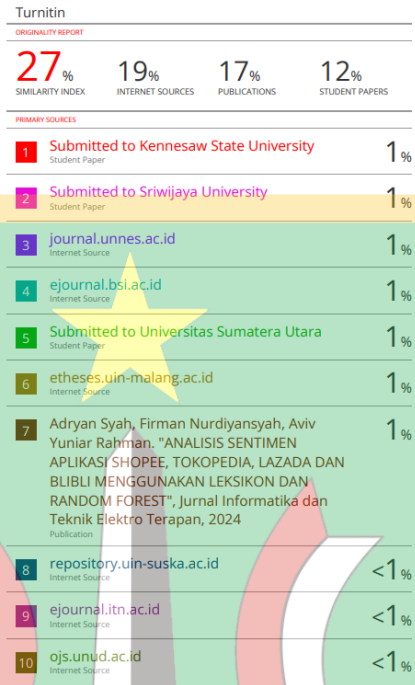
```
st.subheader("Classification Report")
st.write("**Naïve Bayes**")
st.markdown(f"```{classification_report(y_test, y_pred_nb)}```")


st.write("**SVM**")
st.markdown(f"```{classification_report(y_test, y_pred_svm)}```")


st.subheader("Confusion Matrix Algoritma")
fig, axes = plt.subplots(1, 2, figsize=(12, 5))
sns.heatmap(cm_nb, annot=True, fmt='d', cmap='Blues', ax=axes[0])
axes[0].set_title("Naïve Bayes")
sns.heatmap(cm_svm, annot=True, fmt='d', cmap='Greens', ax=axes[1])
axes[1].set_title("SVM")
st.pyplot(fig)
```

4. Dokumentasi Dengan Dosen Pembimbing

5. Hasil Turnitin



Turnitin

ORIGINALITY REPORT

| 27% | 19% | 17% | 12% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Kennesaw State University<br>Student Paper | 1% |
| 2 | Submitted to Sriwijaya University<br>Student Paper | 1% |
| 3 | journal.unnes.ac.id<br>Internet Source | 1% |
| 4 | ejournal.bsi.ac.id<br>Internet Source | 1% |
| 5 | Submitted to Universitas Sumatera Utara<br>Student Paper | 1% |
| 6 | etheses.uin-malang.ac.id<br>Internet Source | 1% |
| 7 | Adryan Syah, Firman Nurdiyansyah, Aviv Yuniar Rahman. "ANALISIS SENTIMEN APLIKASI SHOPEE, TOKOPEDIA, LAZADA DAN BLIBLI MENGGUNAKAN LEKSIKON DAN RANDOM FOREST", Jurnal Informatika dan Teknik Elektro Terapan, 2024<br>Publication | 1% |
| 8 | repository.uin-suska.ac.id<br>Internet Source | <1% |
| 9 | ejournal.itn.ac.id<br>Internet Source | <1% |
| 10 | ojs.unud.ac.id<br>Internet Source | <1% |

6. Bukti Publikasi