

DAFTAR PUSTAKA

- Azhar, R., Surahman, A., & Juliane, C. (2022). Analisis Sentimen Terhadap Cryptocurrency Berbasis Python TextBlob Menggunakan Algoritma Naïve Bayes. In *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 6, Issue 1).
- Bayes Yunitasari, N., Siti Hopipah, H., & Mayasari, R. (2021). Optimasi Backward Elimination untuk Klasifikasi Kepuasan Pelanggan Menggunakan Algoritme k-Nearest Neighbor (k-NN) dan. *Technomedia Journal (TMJ)*, 6(1). <https://doi.org/10.33050/tmj.v6i1>
- Budianto, I., Anwar, S. N., Lomba, J. T., Nomor, J., & Semarang, K. (2022). ANALISIS SENTIMENT PENGGUNA TWITTER MENGENAI PROGRAM VAKSINASI COVID-19 MENGGUNAKAN ALGORITMA NAÏVE BAYES. *Jurnal Teknologi Informasi*, 6(1).
- Djamaludin, M. A., Triayudi, A., & Mardiani, E. (2022). Analisis Sentimen Tweet KRI Nanggala 402 di Twitter menggunakan Metode Naïve Bayes Classifier. *Jurnal Teknologi Informasi Dan Komunikasi*, 6(2), 2022. <https://doi.org/10.35870/jti>
- Fatmawati, F., & Narti, N. (2022). Perbandingan Algoritma C4.5 dan Naive Bayes Dalam Klasifikasi Tingkat Kepuasan Mahasiswa Terhadap Pembelajaran Daring. *JTIM: Jurnal Teknologi Informasi Dan Multimedia*, 4(1), 1–12. <https://doi.org/10.35746/jtim.v4i1.196>
- Febrian, A. K., Chrisnanto, Y. H., Pupita, D., Sabrina, N., & Achmad Yani, J. (2022). SNESTIK Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika Studi Komparasi Metode Klasifikasi K-Nearest Neighbor dan Naïve Bayes dalam Mengidentifikasi Kepuasan Pelanggan Terhadap Produk. *Seminar Nasional Teknik Elektro, Sistem Informasi, Dan Teknik Informatika*, 333. <https://doi.org/10.31284/p.snestik.2022.2717>
- Fitriyah, N., Warsito, B., Asih, D., & Maruddani, I. (2020). ANALISIS SENTIMEN GOJEK PADA MEDIA SOSIAL TWITTER DENGAN KLASIFIKASI SUPPORT VECTOR MACHINE (SVM). *JURNAL GAUSSIAN*, 9(3), 376–390. <https://ejournal3.undip.ac.id/index.php/gaussian/>

- Handoko, W. T., Supriyanto, E., Purwadi, D. I., Budiarmo, Z., & Listiyono, H. (2022). Klasifikasi Opini Pengguna Media Sosial Twitter Terhadap JNT Di Indonesia dengan Algoritma Decision Tree. In *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 6, Issue 2).
- Hendra Kusuma, Y., Suprapto, S., & Setiawan, Y. (2022). *SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat Analysis of Passenger Satisfaction on Airlines Using C4.5 and Naïve Algorithm Analisis Kepuasan Penumpang pada Maskapai Penerbangan Menggunakan Algoritma C4.5 dan Naïve Bayes*. www.kaggle.com
- Hendrik, B., & Suteja, B. R. (2021). Identifikasi Risiko Program Maintenance dalam Pengelolaan Proyek Berbasis Agile Menggunakan Pohon Klasifikasi. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(1). <https://doi.org/10.28932/jutisi.v7i1.3545>
- Herlambang, A. S. D., Komara, E., & Sulisty Herlambang, A. (2021). Pengaruh Kualitas Produk, Kualitas Pelayanan, Dan Kualitas Promosi Terhadap Kepuasan Pelanggan (Studi kasus pada Starbucks Coffee Reserve Plaza Senayan). *Jurnal Ekonomi, Manajemen Dan Perbankan (Journal of Economics, Management and Banking)*, 7(2).
- Jacarria Pangestu, D., & Kodar, A. (2022). *Implementasi Multinomial Naïve Bayes Untuk Klasifikasi Sentimen Terhadap Pelayanan Perusahaan Otobus Menggunakan Data Facebook (Studi Kasus: Grup Facebook Murni Jaya Lovers)*. 7(3).
- Jurnal, A., Rizky Yuliansyah, M., Franz, A., Prodi,), Rekayasa, T., Lunak, P., Dan Informatika, T., Pertanian, P., & Samarinda, N. (2022). Perbandingan Metode K-Nearest Neighbors dan Naïve Bayes Classifier Pada Klasifikasi Status Gizi Balita di Puskesmas Muara Jawa Kota Samarinda. *Adopsi Teknologi Dan Sistem Informasi (ATASI)*, 1. <https://doi.org/10.30872/atasi.v1i1.25>
- Muttaqin Muttaqin, K. S. H. D. L. R. A. P. H. H., & Rahmadini Darwas, T. P. N. N. F. J. K. H. T. H. S. Z. G. D. D. A. A. J. S. (2022). *BIG DATA: Informasi Dalam Dunia Digital* (R. Watrionthos, Ed.). Yayasan Kita Menulis.

- Nuraliza, H., Pratiwi, O. N., & Hamami, F. (2022). Analisis Sentimen IMBd Film Review Dataset Menggunakan Support Vector Machine (SVM) dan Seleksi Feature Importance. In *Jurnal Mirai Manajemen* (Vol. 7, Issue 1).
- Nurhakim, M. A., Widiastiwi, Y., & Chamidah, N. (2022). Analisis Sentimen Terhadap Ulasan Kepuasan Pelanggan Pada Marketplace Tokopedia Di Jejaring Sosial Twitter Menggunakan Algoritma Naïve Bayes.
- PanimalarS, A., ShreeS, V., & KathrineA, V. (2017). The 17 V's Of Big Data. *International Research Journal of Engineering and Technology*. www.irjet.net
- Petiwi, M. I., Triayudi, A., & Sholihati, I. D. (2022). Analisis Sentimen Gofood Berdasarkan Twitter Menggunakan Metode Naïve Bayes dan Support Vector Machine. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 6(1), 542. <https://doi.org/10.30865/mib.v6i1.3530>
- Purwaningsih, E., & Nurelasari, E. (2022). Penerapan Algoritma C4.5 Dalam Pengukuran Tingkat Kepuasan Pelanggan Terhadap Layanan Resto Cepat Saji. *Jurnal Rekayasa Perangkat Lunak*, 3(1). <http://jurnal.bsi.ac.id/index.php/reputasi>
- Rahayu, W. I., Anindita, A., & Fauzan, M. N. (2022). PENENTUAN VALIDASI DATA PEMILIH DAN KLASIFIKASI HASIL PEMILU DPRD KAB.BONE UNTUK MEMREDIKSI PARTAI PEMENANG MENGGUNAKAN METODE NAIVE BAYES Program Studi D4 Teknik Informatika 123 Politeknik Pos Indonesia 123. In *Jurnal Teknik Informatika* (Vol. 14, Issue 1).
- Resti Wardani, N., Saepudin, S., & Warman, C. (2022). Sentimen Analisis Kegiatan Trading Pada Ap-likasi Twitter dengan Algoritma SVM, KNN Dan Random Forrest. In *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 6, Issue 2).
- Rianti, D. L., Umaidah, Y., Voutama, A., & Karawang, S. (2021). Tren Marketplace Berdasarkan Klasifikas Ulasan Pelanggan Menggunakan Perbandingan Kernel Suport Vector Machine. *STRING(Satuan Tulisan Riset Dan Inovasi Teknologi)*, 6(1).
- Rozaqi, A., Triayudi, A., & Aldisa, R. T. (2022). Analisis Sentimen Vaksinasi Booster Berdasarkan Twitter Menggunakan Algoritma Naïve Bayes dan K-NN. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 4(1), 184. <https://doi.org/10.30865/json.v4i1.4907>

- Saurina, N., Rahayuningsih, T., Retnawati, L., Teknik, F., Wijaya, U., & Surabaya, K. (2022). *Analisis Sentimen Ulasan Pelanggan Batik Ecoprint Menggunakan Naïve Bayes Dan KNN Classifier*. 9(2). <http://jurnal.mdp.ac.id>
- Sepri, D., Algoritma, P., Bayes, N., Analisis, U., Penggunaan, K., & Bank, A. (2020). media cetak. *Journal of Computer System and Informatics (JoSYC)*, 2(1), 135–139.
- Simanjuntak, A. D., Buaton, R., & Saragih, R. (2022). DATA MINING KEPUASAN PELANGGAN TERHADAP PELAYANAN PDAM MENGGUNAKAN METODE CLUSTERING. *Jurnal Informatika Kaputama (JIK)*, 6(3).
- Sunday Ade Sitorus, S. E. , M. Si. , CSEM. , CETP. , C. NNLP. , A. P. K. G. S. E. , M. M. , Dr. M. A. S. S. E. , Ak. , M. M. , H. F. S. T. , M. M. , S. S. E. , M. M., & Hesti Umiyati, S. E. , M. M. , P. C. S. MBA. , M. Ed. , CMA. , R. M. G. S. E. , M. M. , M. S. M. Sc. , R. N. A. S. E. , M. M. , Dr. Dra. F. A. M. M. , A. S. Pd. , M. M. , N. T. N. S. E. , M. M. (2022). *Digital Marketing Strategy: Online Marketing Approach* (M. M. A. Sudirman S.E., Ed.). Media Sains Indonesia.
- Susanti, N. A., & Walid, M. (2022a). KLASIFIKASI DATA TWEET UJARAN KEBENCIAN DI MEDIA SOSIAL MENGGUNAKAN NAIVE BAYES CLASSIFIER. In *Jurnal Mahasiswa Teknik Informatika*) (Vol. 6, Issue 2). www.kaggle.com
- Susanti, N. A., & Walid, M. (2022b). KLASIFIKASI DATA TWEET UJARAN KEBENCIAN DI MEDIA SOSIAL MENGGUNAKAN NAIVE BAYES CLASSIFIER. In *Jurnal Mahasiswa Teknik Informatika*) (Vol. 6, Issue 2). www.kaggle.com
- Syafii, A., Dwilestari, G., & Ajiz, A. (2022). KOMPARASI ALGORITMA NAÏVE BAYES DAN ALGORITMA C4.5 DALAM KLASIFIKASI PELANGGAN PRODUK INDIHOME. *JURSIMA Jurnal Sistem Informasi Dan Manajemen* , 10(2). <https://ejournal.stmikgici.ac.id/>
- Tangkelayuk, A., & Mailoa, E. (2022). *Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes Dan Decision Tree*. 9(2), 1109–1119. <http://jurnal.mdp.ac.id>
- Wulandari, A. A., Retno, D., Saputro, S., Program,), Matematika, S., Matematika, F., & Pengetahuan, I. (n.d.). *KLASIFIKASI DATA MINING MENGGUNAKAN NAÏVE*

BAYES CLASSIFIER DENGAN ALGORITMA C5.0 (Classification Data Mining using Naïve Bayes Classifier with C5.0 Algorithm). <https://magestic.unej.ac.id/>





LAMPIRAN

1. SOURCE CODE PYTHON PADA JUPYTER NOTEBOOK.

```
import tweepy
import json
import csv
import pandas as pd
import numpy as np
import re

import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import seaborn as sns
import warnings
import string
import nltk

from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn import tree
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.tree import DecisionTreeClassifier, plot_tree,
export_graphviz
from sklearn.tree import DecisionTreeClassifier
from tweepy import OAuthHandler
from nltk.corpus import stopwords
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.wordnet import WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import TfidfTransformer

from sklearn.metrics import classification_report,
confusion_matrix, accuracy_score
from sklearn.metrics import precision_score, recall_score,
f1_score
```

```

from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split,
cross_val_score
from nltk.stem import PorterStemmer
from IPython.display import display
from textblob import TextBlob
from wordcloud import WordCloud

warnings.filterwarnings('ignore')

CONSUMER_KEY = 'bmjJQ2NqktnmCFE0xwPUe0SdO'
CONSUMER_SECRET =
'Y2EnaMXJIKkP8c9lWIVCiYcadm8OarsjyyrlkNGWuDwtQIexOk'
ACCESS_TOKEN = '1585291357283053569-
2z1wG64LzvBYxpOPsBdprNanlt3mix'
ACCESS_TOKEN_SECRET =
'3yP5aCNsvQI9gukmdoASXht4eo9GmgV2yGUesNja6lS3A'

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
api = tweepy.API(auth, wait_on_rate_limit=True,
wait_on_rate_limit_notify=False, compression=True)

tweets = api.search_tweets(q="starbuck OR starbucks OR
#starbucks")

message, favorite_count, retweet_count, created_at, user_name,
followers_count = [], [], [], [], [], []

for tweet in tweepy.Cursor(api.search_tweets, q="starbuck
OR starbucks OR #starbucks -filter:retweets",

```



```

tweet_mode="extended",    count=10,    result_type="recent",
include_entities=True, lang="id").items(5000):
    message.append(tweet.full_text)
    favorite_count.append(tweet.favorite_count)
    retweet_count.append(tweet.retweet_count)
    created_at.append(tweet.created_at)
    user_name.append(tweet.user.name)
    followers_count.append(tweet.user.followers_count)

list_ = {'Message': message,
        'Favorite Count': favorite_count,
        'Retweet Count': retweet_count,
        'Created At': created_at,
        'Username': user_name,
        'Followers': followers_count}

df = pd.DataFrame(list_)
df.to_csv("CRAWLING.csv")
df

df = pd.read_csv('CRAWLING.csv')
df

import tweepy
import pandas as pd

f=pd.read_csv("CRAWLING.csv")
f

tweet = pd.DataFrame(f["Message"])
tweet

print('Dataset size:',df.shape)
print('Columns are',df.columns)

```

```
df.info()
string.punctuation
df.isnull().sum()

!pip install emoji

from Sastrawi.StopWordRemover.StopWordRemoverFactory
import StopWordRemoverFactory
import matplotlib.pyplot as plt
from nltk.metrics.distance import edit_distance
nltk.download('stopwords')
from Sastrawi.StopWordRemover.StopWordRemoverFactory
import StopWordRemoverFactory
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import emoji
import contractions
import swifter

# CLEANING

def cleaning_text(text):
    # replace RT tag
    text = re.sub('RT\s', '', text)
    # replace @_username
    text = re.sub('\B@\w+', '', text)
    # replace emoji dengan text
    text = emoji.demojize(text)
    # replace URL
    text = re.sub('(\http|https):\/\/\/\S+', '', text)
    # replace #_something_
    text = re.sub('#+', '', text)
```

```

# konversi huruf kapital ke huruf kecil semua
text = text.lower()

# replace kata yang berulang-ulang ('oooooo' menjadi '00')
text = re.sub(r'(\.)\1+', r'\1\1', text)

# replace punctuation repetition dengan single occurrence ('!!!!!!' , menjadi '!')
text = re.sub(r'[\?\.!\!]+(=[\?\.!\!])', ' ', text)

# menghilangkan angka dan spesial karakter, hanya mengambil alfabet saja
text = re.sub(r'^a-zA-Z]', ' ', text)

# replace contractions
text = contractions.fix(text)
return text

df['Message'] = df['Message'].apply(lambda x:
cleaning_text(x))
df.head(10)

# TOKENIZATION

def tokenization(text):
    text = re.split('\W+', text)
    return text

df['Tokenization'] = df['Message'].apply(lambda x:
tokenization(x.lower()))
df.head(10)

from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
kalimat = "ada siapa di mana hehe gagag tata"

```

```
tokens = nltk.tokenize.word_tokenize(kalimat)
kemunculan = nltk.FreqDist(tokens)
print(kemunculan.most_common())

#STOP REMOVAL

stopword = nltk.corpus.stopwords.words('indonesian')

def remove_stopwords(text):
    text = [word for word in text if word not in stopword]
    return text

df['Stop_Removal'] = df['Tokenization'].apply(lambda x:
remove_stopwords(x))
df.head(10)

#CASE FOLDING

df['Message'] = df['Message'].str.lower()
df.head(10)

# STEMMING

# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}
```

```

for document in df['Stop_Removal']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("-----")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term, ":" ,term_dict[term])

print(term_dict)
print("-----")

# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

df['Tweet_Stemmed'] =
df['Stop_Removal'].swifter.apply(get_stemmed_term)
print(df['Tweet_Stemmed'])

from textblob import TextBlob
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()

def stemming_data(x):
    return ps.stem(x)

df['Message'] = df['Message'].apply(stemming_data)

```

```
# LABELLING

dataset = list(df['Message'])
polaritas = 0

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

for i, tweet in enumerate(dataset):
    analysis = TextBlob(tweet)
    polaritas += analysis.polarity

    if analysis.sentiment.polarity > 0:
        total_positif += 1
        status.append('positive')
    elif analysis.sentiment.polarity == 0:
        total_netral += 1
        status.append('neutral')
    else:
        total_negatif += 1
        status.append('negative')

    total += 1

print(f'Hasil Analisis Data:\nPositif = {total_positif}\nNetral = {total_netral}\nNegatif = {total_negatif}')
print(f'\nTotal Data : {total}')

status = pd.DataFrame({'klasifikasi': status})
df['klasifikasi'] = status
df
```

```

df.to_csv("CRAWLING TO PREPROCESSING.csv")
df = pd.read_csv('CRAWLING TO PREPROCESSING.csv')
df

# Melihat jumlah baris dan kolom

print('Dataset size:',df.shape)
print('Columns are',df.columns)

# Tipe data setiap kolom

df.info()

ANALIZI CHART_KLASIFIKASI SENTIMENT
# VALIDASI

def show_pie(label, data, legend_title) :
    fig, ax = plt.subplots(figsize=(8, 10),
subplot_kw=dict(aspect='equal'))

    labels = [x.split()[-1] for x in label]

    def func(pct, allvals):
        absolute = int(pct/100.*np.sum(allvals))
        return "{:.1f}% ({:d})".format(pct, absolute)

    wedges, texts, autotexts = ax.pie(data, autopct=lambda
pct: func(pct, data),

textprops=dict(color="w"))

    ax.legend(wedges, labels,
              title= legend_title,

```

```

        loc="center left",
        bbox_to_anchor=(1, 0, 0.5, 1))

plt.setp(autotexts, size=10, weight="bold")
plt.show()

from wordcloud import WordCloud, STOPWORDS

def plot_cloud(wordcloud):
    plt.figure(figsize=(12, 8))
    plt.imshow(wordcloud)
    plt.axis("off");

all_words = ' '.join([tweets for tweets in df['Message']])
wordcloud = WordCloud(width = 3000, height = 2000,
random_state=3, background_color='white', colormap='Set2',
collocations=False, stopwords =
STOPWORDS).generate(all_words)
plot_cloud(wordcloud)

TfidfTransformer

df = pd.read_csv('CRAWLING TO PREPROCESSING.csv')
df.tail(20)

# replace label pada dataset
df.klasifikasi.replace("neutral", 1 , inplace = True)
df.klasifikasi.replace("positive", 2 , inplace = True)
df.klasifikasi.replace("negative", 0 , inplace = True)
df.head(10)

from sklearn.feature_extraction.text import
TfidfTransformer

```



```

tf_idf_vectorizer =
TfidfVectorizer(use_idf=True,ngram_range=(1,3))
final_vectorized_data =
tf_idf_vectorizer.fit_transform(df['Tweet_Stemmed'])

final_vectorized_data

SPLIT DATA TESTING DAN TRAINING
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(final_vectorized_data, df['klasifikasi'],

test_size=0.8, random_state=50)

print("X_train_shape : ",X_train.shape)
print("X_test_shape : ",X_test.shape)
print("y_train_shape : ",y_train.shape)
print("y_test_shape : ",y_test.shape)

KLASIFIKASI NAIVE BAYES

from sklearn.naive_bayes import MultinomialNB # Naive Bayes
Classifier

model_naive = MultinomialNB().fit(X_train, y_train)
predicted_naive = model_naive.predict(X_test)

# Evaluation metrics
from sklearn.metrics import confusion_matrix
conn1 = confusion_matrix(y_test, predicted_naive)
conn1

from sklearn.metrics import confusion_matrix

```

```

plt.figure(dpi=600)
mat = confusion_matrix(y_test, predicted_naive)
sns.heatmap(mat.T, annot=True, fmt='d', cbar=False)

plt.title('Confusion Matrix for Naive Bayes')
plt.xlabel('true label')
plt.ylabel('predicted label')
# plt.savefig("gdrive/My
Drive/projekml/assets/confusion_matrix.png")
plt.show()

from sklearn.metrics import accuracy_score

score_naive = accuracy_score(predicted_naive, y_test)
print("Accuracy dengan Naive-bayes: ",score_naive)

from sklearn.metrics import classification_report
print(classification_report(y_test, predicted_naive))

KLASIFIKASI DECISION TREE

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier()
dtree.fit(X_train,y_train)
predicted_dtree = dtree.predict(X_test)

# Evaluation metrics
from sklearn.metrics import confusion_matrix
conn1 = confusion_matrix(y_test, predicted_dtree)
conn1

from sklearn.metrics import confusion_matrix

```

```

plt.figure(dpi=600)
mat = confusion_matrix(y_test, predicted_dtree)
sns.heatmap(mat.T, annot=True, fmt='d', cbar=False)

plt.title('Confusion Matrix for Decision Tree')
plt.xlabel('true label')
plt.ylabel('predicted label')
# plt.savefig("gdrive/My
Drive/projekml/assets/confusion_matrix_DT.png")
plt.show()

from sklearn.metrics import accuracy_score

score_dtree= accuracy_score(predicted_dtree, y_test)
print("Accuracy dengan Decision Tree: ",score_dtree)

from sklearn.metrics import classification_report
print(classification_report(y_test, predicted_dtree))

```

2. SOURCE CODE UNTUK MEMBUAT TAMPILAN STREAMLIT.

```

import streamlit as st
import numpy as np
import pandas as pd
import csv
import matplotlib.pyplot as plt
import altair as alt
import pickle
import plotly.express as px
from datetime import datetime, date
from PIL import Image
from sklearn import model_selection

```

```

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import
TfidfVectorizer

from sklearn.metrics import accuracy_score
import seaborn as sns; sns.set(font_scale=1.2)
from sklearn.svm import LinearSVC

n_bar = st.sidebar.radio("Navigation", ["Home", "Dataset",
"Naive Bayes Classifier", "Decision Tree Classifier",
"Visualisasi"])

#home section
if n_bar == "Home":
    st.markdown("<h1 style='text-align:
center;'>IMPLEMENTASI ALGORITMA DECISION TREE & NAIVE
BAYES UNTUK KLASIFIKASI SENTIMEN TERHADAP KEPUASAN
PELANGGAN STARBUCKS MENGGUNAKAN TWITTER API</h1>",
unsafe_allow_html=True)
    col1, col2, col3= st.columns(3)

    with col1:
        st.write(' ')

    with col2:
        st.image('STARBUCKS.jpg')

    with col3:
        st.write(' ')

# Dataset section
if n_bar == "Dataset":

```

```

st.title('📍 Starbucks Dataset')

st.markdown('''

    Proses pengumpulan data ( Crawling Data ), Pre-
    processing, serta Validasi dan Evalusi menggunakan bahasa
    pemrograman python yang terdapat pada IDE Jupyter Notebook
    dimana hasil data yang diperoleh akan disimpan menggunakan
    format file csv .

    ''')

# isExpand = st.checkbox('Expand all questions')

with st.expander('CRAWLING DATA ', expanded=True):
    st.write('Crawling Data merupakan tahapan awal
    pada penelitian ini, dimana prosedur crawling data
    menggunakan twitter API dengan bahasa pemrograman python,
    lalu mencari data tweet dengan kata kunci "starbucks OR
    STARBUCKS OR #STARBUCKS.')

    st.subheader('Berikut ini hasil dari Crawling
    Data.')

    myData = pd.read_csv('CRAWLING.csv')

    st.dataframe(myData)

    st.write('Didapatkan Dataset sebanyak 2478
    Tweet.')

with st.expander('PRE-PROCESSING', expanded=True):
    st.write('Pre-processing - dimana diperlukan untuk
    membersihkan data yang bertujuan agar menghilangkan noisy
    terhadap data supaya hasil perhitungan data tersebut

```

menjadi lebih akurat dan dapat digunakan pada tahapan selanjutnya.')

```
st.subheader('Berikut ini hasil dari Pre-  
processing Data.')
```

```
PREPROCESSING = pd.read_csv('CRAWLING TO  
PREPROCESSING.csv')
```

```
st.dataframe(PREPROCESSING)
```

```
st.write('1. Cleaning melakukan penghapusan kata  
pada teks yang tidak memiliki arti, seperti kata  
penghubung, tanda baca, simbol atau karakter, angka, emoji  
dan URL.')
```

```
st.write('2. Tokenizing dimana pada proses ini  
bertugas untuk menyederhakan isi pada text yang awalnya  
berupa kalimat menjadi kata terpisah.')
```

```
st.write('3. Stopword Removal dimana bertugas  
untuk melakukan penghapusan kata yang tidak penting  
contohnya yakni kata sambung dan kata ganti orang. Dari  
proses-proses tersebut akan menghasilkan dataset baru.')
```

```
st.write('4. case folding dimana pada tahapan ini  
bertugas untuk melakukan perubahan huruf kapital yang  
terdapat pada text menjadi kecil.')
```

```
st.write('5. Stemming dimana pada tahapan ini  
bertugas untuk mencari akar kata dengan menghilangkan  
imbuhan pada sebuah kata. .')
```

```
st.write('6. Labelling dimana data yang dihasilkan  
dari proses Stopword Removal akan dilakukan perhitungan  
berdasarkan polarity dari ulasan opini yang didapatkan,  
sehingga memperoleh 3 kategori pada klasifikasi yakni,  
label positif.')
```

```
#Naive Bayes Classifier
```

```
if n_bar == "Naive Bayes Classifier":
```

```
st.title("🇮🇩 Naive Bayes Classifier")

st.subheader("Evaluasi kinerja matriks yang akan  
digunakan: ")

st.write('Klasifikasi sentimen analis dengan algoritma  
Naïve Bayes pada penelitian ini menggunakan bahasa  
pemrograman python yang terdapat pada jupyter notebok,  
dimana mendapatkan hasil yakni : laporan klasifikasi  
(classification report), penarikan kembalikan data (recall),  
presisi (precision), dan hasil akurasi (accuracy). Pada  
uji dataset dengan algoritma Naïve Bayes digambarkan  
berupa Confusion Matrix, dimana dari gambaran tersebut  
didapatkan hasil komparasi klasifikasi pada sistem yakni  
hasil klasifikasi sebenarnya.')

col1, col2 = st.columns(2)
with col1:
    st.subheader('1. Confusion Matrix ')

with col2:
    st.write('')

col3, col4 = st.columns(2)
with col3:
    st.image('CM.NB.png')

with col4:
    st.write('Pada gambar disamping menjelaskan dimana  
Confusion Matrix tersebut berbentuk matriks dengan  
ukuran 3 x 3 yang mengkategorikan masing-masing kelas  
klasifikasi menjadi Neutral untuk angka 1, positif untuk  
angka 2 dan negative untuk angka 0.')

st.write('')
```

```
st.markdown('''
```

Hasil Confusion matrix dari masing-masing kelas yang terdapat pada kondisi kelas sebenarnya (actual class), dimana dapat memprediksi model dengan benar dan dapat dijelaskan sebagai berikut, nilai TP sebanyak 30 data untuk kelas negative, nilai TP sebanyak 1404 data untuk kelas neutral dan nilai TP sebanyak 33 data untuk kelas positive. Dalam melakukan evaluasi pada matriks untuk mendapatkan perhitungan nilai akurasi dengan algoritma Naïve Bayes menggunakan bahasa pemrograman python yang terdapat pada jupyter notebook dan dapat dilihat pada gambar sebagai berikut:

```
#
'''
st.subheader('2. Precision, Recall & F1-Score')
st.write(''''
support | precision | recall | f1-score |
-----|-----|-----|-----|
0 | 0.97 | 0.15 | 0.26 |
1 | 0.73 | 1.00 | 0.84 |
2 | 1.00 | 0.09 | 0.16 |
accuracy | 0.74 |
1983 |
macro avg | 1.00 | 0.41 | 0.42 |
1983 |
weighted avg | 0.81 | 0.74 | 0.66 |
1983 |

''')
```



```
st.write('')
```

```
st.write('')
```

```
st.write(''
```

Gambar diatas merupakan hasil perhitungan dari matriks menggunakan bahasa pemrograman python yang terdapat pada jupyter notebok yakni 0,74 atau 74%.

Berikut terdapat penjelasan perhitungan terhadap nilai akurasi secara manual dari perhitungan matriks : \n

```
```Python
```

```
Akurasi = (True Positif+True Negatif)/(Total data
matrix) x 100%\n
```

```
 = 1467/1983 x 100%\n
```

```
 = 74%
```

```
```\n
```

```
''')
```

```
#Decision Tree Classifier
```

```
if n_bar == "Decision Tree Classifier":
```

```
    st.title("🇮🇩 Decision Tree Classifier")
```

```
    st.subheader("Evaluasi kinerja matriks yang akan  
digunakan: ")
```

st.write('Klasifikasi sentimen analis dengan algoritma Naïve Bayes pada penelitian ini menggunakan bahasa pemrograman python yang terdapat pada jupyter notebok, dimana mendapatkan hasil yakni : laporan klasifikasi (classification report), penarikan kermali data (recall), presisi (precision), dan hasil akurasi (accuracy). Pada uji dataset dengan algoritma Naïve Bayes digambarkan berupa Confusion Matrix, dimana dari gambaran tersebut

didapatkan hasil komparasi klasifikasi pada sistem yakni hasil klasifikasi sebenarnya.')

```
col1, col2 = st.columns(2)
```

```
with col1:
```

```
    st.subheader('1. Confusion Matrix ')
```

```
with col2:
```

```
    st.write('')
```

```
col3, col4 = st.columns(2)
```

```
with col3:
```

```
    st.image('C.M.DT.png')
```

```
with col4:
```

st.write('Pada gambar disamping menjelaskan dimana Confusion Matrix tersebut berbentuk matriks dengan ukuran 3 x 3 yang mengkategorikan masing-masing kelas klasifikasi menjadi Neutral untuk angka 1, positif untuk angka 2 dan negative untuk angka 0.')

```
    st.write('')
```

```
st.write('')
```

```
####
```

Hasil Confusion matrix dari masing-masing kelas yang terdapat pada kondisi kelas sebenarnya (actual class), dimana dapat memprediksi model dengan benar dan dapat dijelaskan sebagai berikut, nilai TP sebanyak 126 data untuk kelas negatif, nilai TP sebanyak 1272 data untuk kelas neutral dan nilai TP sebanyak 252 data untuk kelas positif. Dalam melakukan evaluasi pada matriks untuk mendapatkan perhitungan nilai akurasi dengan algoritma Decision Tree menggunakan bahasa pemrograman python yang terdapat pada jupyter notebok dan dapat dilihat pada gambar sebagai berikut:

```
''')
```

```

st.subheader('2. Precision, Recall & F1-Score')
st.write('')

|          | precision | recall | f1-score |
support |
----- |
|          |          |          |          |
201      |          |          |          |
|          |          |          |          |
1405     |          |          |          |
|          |          |          |          |
377      |          |          |          |
|          |          |          |          |
1983     | accuracy |          |          |
|          |          |          |          |
1983     | macro avg |          |          |
|          |          |          |          |
1983     | weighted avg |          |          |

'''

st.write('')
st.write('')

st.write('')

```

		precision	recall	f1-score
201	0	0.61	0.63	0.62
1405	1	0.88	0.91	0.89
377	2	0.75	0.67	0.71
1983	accuracy			0.83
1983	macro avg	0.75	0.73	0.74
1983	weighted avg	0.83	0.83	0.83

Gambar diatas merupakan hasil perhitungan dari matriks menggunakan bahasa pemrograman python yang terdapat pada jupyter notebok yakni 0,83 atau 83%.

Berikut terdapat penjelasan perhitungan terhadap nilai akurasi secara manual dari perhitungan matriks :\n

```Python

**Akurasi** = (True Positif+True Negatif)/(Total data matrix) x 100% \n

```

 = 1650/1983 x 100%\n
 = 83%.

 ...

 '''

#Visualisasi section
if n_bar == "Visualisasi":

 st.title("📊 Validasi")

 st.write('Pada tahapan validasi terdapat proses dengan
kegunaan untuk melakukan klasifikasi sentimen kepuasan
pelanggan terkait penerapan algoritma Decision tree dan
Naïve Bayes, serta dilakukan proses yang akan memperoleh
nilai prediksi kelas kepuasan pelanggan atau label terkait
data testing dan data training. ')

 st.subheader("Validasi kinerja metrik yang didapatkan
:")

 col1, col2 = st.columns(2)
 with col1:

 st.subheader('1. Pie Chart ')

 with col2:

 st.write('')

 col3, col4 = st.columns(2)
 with col3:

 st.image('PIE.png')

 with col4:

```

```
st.write('Pada proses ini dimana data yang dihasilkan dari proses Stopword Removal akan dilakukan perhitungan berdasarkan polarity dari ulasan opini yang didapatkan, sehingga memperoleh 3 kategori pada klasifikasi yakni, label negatif dimana (0) dan label yang bernilai netral (nilai 1, sedangkan untuk) label positif dimana (nilai 2).')
```

```
st.write(''
```

```
####
```

Pada Pie Chart diatas dapat dijelaskan dimana klasifikasi ulasan opini masyarakat terhadap starbucks menggunakan jejaring media sosial twitter, dapat dilihat label yang bernilai positif sebesar 19,2%, label yang bernilai negatif sebesar 10,4% dan label yang bernilai netral sebesar 70,3%.

```
''')
```

```
col5, col6 = st.columns(2)
```

```
with col5:
```

```
st.subheader('2. Wordcloud')
```

```
with col6:
```

```
st.write('')
```

```
col7, col8 = st.columns(2)
```

```
with col7:
```

```
st.image('wordcloud.png')
```

```
with col8:
```

```
st.write('')
```

```
st.write(''

''')
```



TIYAS ASIH QURNIA PUTRI - 197006416012 - SISTEM  
INFORMASI

ORIGINALITY REPORT

26%

SIMILARITY INDEX

24%

INTERNET SOURCES

9%

PUBLICATIONS

8%

STUDENT PAPERS

PRIMARY SOURCES

|   |                                                                                                     |    |
|---|-----------------------------------------------------------------------------------------------------|----|
| 1 | <a href="http://journal.sekawan-org.id">journal.sekawan-org.id</a><br>Internet Source               | 2% |
| 2 | <a href="http://pdfs.semanticscholar.org">pdfs.semanticscholar.org</a><br>Internet Source           | 2% |
| 3 | <a href="http://tunasbangsa.ac.id">tunasbangsa.ac.id</a><br>Internet Source                         | 2% |
| 4 | <a href="http://ejournal.stmikgici.ac.id">ejournal.stmikgici.ac.id</a><br>Internet Source           | 1% |
| 5 | <a href="http://jurnal.bsi.ac.id">jurnal.bsi.ac.id</a><br>Internet Source                           | 1% |
| 6 | <a href="http://ejurnal.stmik-budidarma.ac.id">ejurnal.stmik-budidarma.ac.id</a><br>Internet Source | 1% |
| 7 | <a href="http://pastebin.com">pastebin.com</a><br>Internet Source                                   | 1% |
| 8 | <a href="http://www.researchgate.net">www.researchgate.net</a><br>Internet Source                   | 1% |
| 9 | <a href="http://jurnal.kaputama.ac.id">jurnal.kaputama.ac.id</a><br>Internet Source                 | 1% |