LAMPIRAN

Lampiran 1. 1 CV Expert Labelling

Irfan Abdul Hakim / +62 821-3535-7602/ hirfan825@gmail.com

SUMMARY

A young professional who is interested in social projects, corporate social responsibilities, social research, learning development, and education. Highly articulative and creative with strong interpersonal communication. Experienced in education, observing, Human resource and interviewing.

EDUCATION

Bachelor of Sociology (S. Sos) | University of Gajah Mada | **GPA:** 3.40 (out of 4.00)

Master of Arts (M. A.) | University of Gadjah Mada| GPA: 3.50 (out of 4.00)

WORK EXPERIENCE

OFFICE STAFF

Panitia Pengawas Pemilu (Panwaslu) Kecamatan Semboro | 2022

- Report election violations
- Documenting administration report about election violations

TEACHER

MTs Ali Maksum Yogyakarta | 2021

SMA Nurul Muslim Batealit Jepara| 2023

- Collecting and analyzing data, providing statistical reports, and interview
- Microteaching and observation
- Designed effective teaching tools for learning development

RESEARCH ASSISTANT

Pusat Studi Pancasila Universitas Pembangunan Nasional Yogyakarta | 2019

Collecting and analyzing data, providing statistical reports, and interview

PROJECT EXPERIENCE

MODERATOR

National Seminar Entrepreneur of PMII Hasyim Asy'arie UNY| 2020

LOGISTIC

Acceptance of new students MA Ali Maksum Krapyak Yogyakarta | 2020

COORDINATOR ENUMERATOR

Customer Satisfaction Survey in PDAM Yogyakarta | 2020

FINANCE MANAGER

Children's Party | 2020

LOGISTIC

Farewell Party of Sosiology | 2020

Irfan Abdul Hakim / +62 821-3535-7602/ hirfan825@gmail.com

PROFESSIONAL SKILL

RESEARCH SKILL

- Designed effective teaching tools
- Microteaching
- observation
- interview
- statistical analysis

PERSONAL TRAIT

- Creative
- Fast Learner
- Team Player
- Highly Motivated
- Adaptive

LANGUAGE

- Indonesian | **Native**
- English | **Professional Working Proficiency**

AWARD

- Finalist essay's competition of Jala PRT| **2022**

RESEARCH& PUBLICATION

- Strategi Dakwah Komunitas Arus Informasi Santri Nusantara| **2020**

ORGANIZATION

- Gerakan Mahasiswa Satu Bangsa (GEMASABA) Kab. Sleman | Vice Chairman | **2021 – 2022,**
- Dormitory Administrator of MTs Ali Maksum Pondok Pesantren Krapyak Yogyakarta | Chief | **2015-2022**
- Ikatan Alumni MA Ali Maksum Yogyakarta | Chief | **2017**
- Pergerakan Mahasiswa Islam Indonesia (PMII) Gadjah Mada | Manager of Caderitation |

Lampiran 2. 1 Code Crawling Twitter

```
#@title Twitter Auth Token

twitter_auth_token = '************************' # change this auth
token
```

```
# Import required Python package
!pip install pandas

# Install Node.js (because tweet-harvest built using Node.js)
!sudo apt-get update
!sudo apt-get install -y ca-certificates curl gnupg
!sudo mkdir -p /etc/apt/keyrings
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key
| sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg
```

```
!NODE_MAJOR=20 && echo "deb [signed-
by=/etc/apt/keyrings/nodesource.gpg]
https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | sudo
tee /etc/apt/sources.list.d/nodesource.list

!sudo apt-get update
!sudo apt-get install nodejs -y

!node -v
```

```
# Crawl Data

filename = 'dataset.csv'
search_keyword = 'Bitcoin Halving lang:id'
limit = 600

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --
tab "LATEST" -l {limit} --token {twitter_auth_token}
```

```
import pandas as pd

# Specify the path to your CSV file
file_path = f"tweets-data/{filename}"

# Read the CSV file into a pandas DataFrame
df = pd.read_csv(file_path, delimiter=",")

# Display the DataFrame
display(df)
```

```
# Cek jumlah data yang didapatkan

num_tweets = len(df)
print(f"Jumlah tweet dalam dataframe adalah {num_tweets}.")
```

Lampiran 2. 2 Import Library & Pip Install

```
!pip install Sastrawi
import pandas as pd
import numpy as np
import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report


# Download data NLTK
nltk.download('punkt')
nltk.download('stopwords')
```

Lampiran 2. 3 Read Dataset & Count Sentiment

```
df = pd.read_csv('dataset_uji.csv')

# jumlah sentimen
sentimen_counts = df['Sentimen'].value_counts()
print(sentimen_counts)

for sentimen, count in sentimen_counts.items():
    print(f"Jumlah Sentimen {sentimen}: {count}")
```

Lampiran 2. 4 Preprocessing

```
# Load dataset
file_path = 'dataset_uji.csv'
data = pd.read_csv(file_path)

# Mengganti NaN dengan string kosong
data['full_text'] = data['full_text'].fillna('')

# Inisialisasi stemmer untuk Bahasa Indonesia
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# Fungsi untuk case folding
def case_folding(text):
    return text.lower()

# Fungsi untuk cleansing
def cleansing(text):
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'@\w+|#\w+', '', text)
    text = re.sub(r'[^a-z\s]', '', text)
    return text

# Fungsi untuk tokenizing
def tokenizing(text):
    return word_tokenize(text)
```

```
# Fungsi untuk stopword removal
def stopword_removal(tokens):
    stop_words = set(stopwords.words('indonesian'))
    return [word for word in tokens if word not in stop_words]

# Fungsi untuk stemming
def stemming(tokens):
    return ' '.join([stemmer.stem(word) for word in tokens])

# Preprocessing teks
data['case_folding'] = data['full_text'].apply(case_folding)
data['cleansing'] = data['case_folding'].apply(cleansing)
data['tokenizing'] = data['cleansing'].apply(tokenizing)
data['stopword_removal'] = data['tokenizing'].apply(stopword_removal)
data['stemming'] = data['stopword_removal'].apply(stemming)

# Tampilkan hasil preprocessing
data[['full_text', 'case_folding', 'cleansing', 'tokenizing',
'stopword_removal', 'stemming', 'Sentimen']].head()
```

Lampiran 2. 5 Code Untuk Menyimpan Hasil Teks Preprocessing

```
output_path = 'preprocessed_text.csv'
columns_to_save = ['full_text', 'case_folding', 'cleansing',
'tokenizing', 'stopword_removal', 'stemming', 'Sentimen']
data[columns_to_save].to_csv(output_path, index=False)
print(f"Data hasil preprocessing telah disimpan di: {output_path}")

# Menampilkan hasil preprocessing
data[['full_text', 'case_folding', 'cleansing', 'tokenizing',
'stopword_removal', 'stemming', 'Sentimen']].head()
```

Lampiran 2. 6 Code Delete Duplicate

```
df_preprocessed = pd.read_csv('preprocessed_text.csv')

df_preprocessed['stemming'] = df_preprocessed['stemming'].fillna(' ')
df_preprocessed.dropna(subset=['stemming'], inplace=True)
df_preprocessed.drop_duplicates(subset=['stemming'], inplace=True)

# Menyimpan data yang telah dibersihkan ke dalam file baru
df_preprocessed.to_csv('cleaned_text_no_duplicates.csv', index=False)

# Menampilkan data yang telah dihapus duplikatnya
print("Data setelah duplikat dihapus:")
```

```
print(df_preprocessed)
```

Lampiran 2. 7 Code Visualisasi Persentase Sentimen

```
df = pd.read_csv('cleaned_text_no_duplicates.csv')
sentimen_counts = df['Sentimen'].value_counts()

# pie chart
plt.figure(figsize=(8, 8))
plt.pie(sentimen_counts, labels=sentimen_counts.index,
autopct='%1.1f%%', startangle=140, colors=['#ff9999','#66b3ff'])
plt.axis('equal')
plt.show()
```

Lampiran 2. 8 Code Wordcloud Sebelum Teks Preprocessing

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
df = pd.read_csv('dataset.csv')

text = " ".join(review for review in df['full_text'])
wordcloud = WordCloud(width=800, height=400, background_color='white',
stopwords={'bitcoin', 'halving'}).generate(text)

# Menampilkan WordCloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Lampiran 2. 9 Code Wordcloud Setelah Teks Preprocessing

```
df = pd.read_csv('cleaned_text_no_duplicates.csv')

bitcoin_halving_texts = '
'.join(df[df['stemming'].str.contains('bitcoin halving',
na=False)]['stemming'].dropna())

# Define stopwords
stopwords = set(STOPWORDS)
stopwords.update(['bitcoin', 'halving', 'bitcoin halving'])
wordcloud = WordCloud(width=800, height=400, background_color='white',
stopwords=stopwords).generate(bitcoin_halving_texts)

# Display the WordCloud
```

```
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Lampiran 2. 10 TF-IDF (Term Frequency – Inverse Document Frequency)

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

df = pd.read_csv('cleaned_text_duplicates.csv')
tfidf_vectorizer = TfidfVectorizer()

tfidf_matrix = tfidf_vectorizer.fit_transform(df['stemming'])

sample_index = 1
sample_text = df['stemming'].iloc[sample_index]
tfidf_sample_matrix = tfidf_vectorizer.transform([sample_text])

terms = tfidf_vectorizer.get_feature_names_out()
tfidf_values = tfidf_sample_matrix.toarray()[0]

idf_values = tfidf_vectorizer.idf_
tf_values = (tfidf_sample_matrix > 0).astype(int).toarray()[0] /
len(sample_text.split())
tfidf_df = pd.DataFrame({
    'TF': tf_values,
    'IDF': idf_values,
    'TF-IDF': tfidf_values,
    'Term': terms
})

tfidf_df = tfidf_df[tfidf_df['TF-IDF'] > 0]
tfidf_df = tfidf_df.sort_values(by='TF-IDF', ascending=False)

print(f"Show TFIDF sample ke-{sample_index}")
print(tfidf_df)
print(f"\nTFIDF Table for Sample Index {sample_index}:\n")
print(f"{'Array Position':<20}{'TF':<10}{'IDF':<10}{'TF-
IDF':<10}{'Term':<20}")
print("="*60)
for i, row in tfidf_df.iterrows():
    print(f"{i:<20}{row['TF']:<10.6f}{row['IDF']:<10.6f}{row['TF-
IDF']:<10.6f}{row['Term']:<20}")
```

Lampiran 2. 11 Naive Bayes Classification

```
def evaluate_naive_bayes(df, test_size):
    X = df['stemming']
    y = df['Sentimen']
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=test_size, random_state=42)

    vectorizer = TfidfVectorizer(max_features=1000, min_df=5,
max_df=0.7)
    X_train_tfidf = vectorizer.fit_transform(X_train)
    X_test_tfidf = vectorizer.transform(X_test)

    naive_bayes = MultinomialNB(alpha=0.1)
    naive_bayes.fit(X_train_tfidf, y_train)

    y_pred = naive_bayes.predict(X_test_tfidf)

    accuracy = accuracy_score(y_test, y_pred)
    print(f'Naive Bayyes Accuracy {1-test_size:.0%}:{test_size:.0%}:
{accuracy:.4f}')
    print(classification_report(y_test, y_pred, zero_division=1))

ratios = [0.1, 0.2, 0.3]
accuracies = []

for ratio in ratios:
    accuracy = evaluate_naive_bayes(df, test_size=ratio)
    accuracies.append(accuracy)
```

Lampiran 2. 12 Code Confusion Matrix

```
 # Hitung confusion matrix
    cm = confusion_matrix(y_test, y_pred)

    # Plot confusion matrix
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=['Negatif', 'Positif'], yticklabels=['Negatif',
'Positif'])
    plt.xlabel('Prediksi')
    plt.ylabel('Aktual')
    plt.title(f'Confusion Matrix {1-test_size:.0%}:{test_size:.0%}')
    plt.show()
```

Lampiran 3. 1 Kartu Kendali Bimbingan

KARTU KENDALI BIMBINGAN LAPORAN KARYA ILMIAH

Nama	:	Andi Nur Halim
NIM	:	2011102441038
Nama Dosen Pembimbing	:	Rudiman, S.Kom., M.Sc
Judul Penelitian	:	Analisis Sentimen Opini Publik Terhadap Peristiwa Bitcoin Halving Pada Data Teks Twitter Menggunakan Metode Naïve Bayes Dan Pembobotan Fitur TF-IDF

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1	7/2/2024	Persetujuan bimbingan dengan dosen	
2	14/2/2024	Mencari topik permasalahan yang akan digunakan sebagai objek penelitian.	
3	22/2/2024	Evaluasi objek penelitian	
4	29/2/2024	Menentukan Judul penelitian dan Latar belakang	
5	9/3/2024	melakukan penulisan Latar belakang masalah serta Judul dan arahan dosen	
6	13/03/2024	Revisi penulisan Latar belakang	
7	18/3/2024	Memberikan arahan dalam penulisan canvas pengajuan Judul.	
8	27/3/2024	Memperbaiki revisi dan saran di bab 1-2	
9	5/4/2024	Revisi penulisan bab 2	
10	29/4/2024	Menulis penulisan bab 3 dan membuat code TF-IDF serta arahan	
11	26/5/2024	dosen pembimbing memberikan masukan di bab 3	
12	17/07/2024	Memberikan revisi menyelesaikan Jurnal dan Skripsi	
13	30/07/224	Melakukan revisi jurnal yang akan di submit oleh dosen pembimbing	

Mengetahui

Dosen Pembimbing

Rudiman, S.Kom., M.Sc
NIDN. 1105068202

Ketua Program Studi

Arbansyah, S.Kom., M.Ti
NIDN. 1119019203

SKRIPSI ANDI NUR HALIM

by Teknik Informatika UMKT

Submission date: 25-Jul-2024 09:24AM (UTC+0800)
Submission ID: 2422037094
File name: SKRIPSI_ANDI_NUR_HALIM.docx (1.03M)
Word count: 6837
Character count: 44291

SKRIPSI ANDI NUR HALIM

ORIGINALITY REPORT

19%	16%	11%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ojs.cahayamandalika.com Internet Source	1%
2	Submitted to Universitas Negeri Semarang - iTh Student Paper	1%
3	ejournal.itn.ac.id Internet Source	1%
4	jurnalsyntaxadmiration.com Internet Source	1%
5	www.ejurnal.stmik-budidarma.ac.id Internet Source	1%
6	dspace.umkt.ac.id Internet Source	1%
7	Almi yulistia Alwanda, Ema Utami, Ainul Yaqin. "Analisis Klasifikasi Konsentrasi Mahasiswa Menggunakan Algoritma K-Nearest Neighbor", Infotek: Jurnal Informatika dan Teknologi, 2024 Publication	1%