

LAMPIRAN

L1 CV Expert Labelling Data

Irfan Abdul Hakim

Hirfan825@gmail.com/082135357602

PROFIL

Saya adalah seorang profesional yang memiliki keterampilan komunikasi yang baik, pemikiran terbuka, dan selalu bersemangat dalam mencari pengetahuan baru. Saya memiliki keinginan yang besar untuk terus belajar dan berkembang. Saya dengan cepat dapat menyesuaikan diri dengan perubahan dan memiliki kemampuan untuk beradaptasi dengan cepat dalam lingkungan kerja yang dinamis. Saya juga sangat berpengetahuan dalam penggunaan teknologi dan memiliki kemampuan untuk memanfaatkannya secara efektif. Saya mencari peluang untuk bekerja dalam lingkungan yang inovatif, di mana saya dapat menerapkan keterampilan analitis dan berpikir kritis saya untuk mencapai hasil yang optimal.

PENDIDIKAN

UNIVERSITAS GADJAH MADA

2015-2020

S1 Sosiologi, IPK: 3.40/4.00

S2 Sosiologi, IPK: 3.50/ 4.00

PENGALAMAN KERJA

SMA NURUL MUSLIM BATEALIT JEPARA

STAF PENGAJAR SOSIOLOGI

- Merancang Silabus pembelajaran di Awal Semester.
- Melakukan aktifitas belajar mengajar sesuai dengan rancangan awal silabus.
- Melakukan evaluasi hasil pembelajaran melalui penilaian harian, penilaian tengah semester dan penilaian akhir semester.

PANITIA PENGAWAS PEMILU (PANWASLU) KECAMATAN SEMBORO

2022 - sekarang

STAF PELAKSANA

- Membantu Tugas Pengawasan komisioner panwaslu kecamatan.
- Melakukan pendataan laporan dan temuan pelanggaran baik dari aduan masyarakat maupun temuan langsung Panwaslu.
- Melakukan Pengarsipan Surat Keluar Masuk Laporan dan Temuan Pelanggaran.

MTS ALI MAKSUM YOGYAKARTA

2021 - 2022

STAF PENGAJAR

- Merancang Silabus pembelajaran di Awal Semester.
- Melakukan aktifitas belajar mengajar sesuai dengan rancangan awal silabus.
- Melakukan evaluasi hasil pembelajaran melalui penilaian harian, penilaian tengah semester dan penilaian akhir semester.

PENGALAMAN ORGANISASI

GERAKAN MAHASISWA SATU BANGSA (GEMASABA) KAB. SLEMAN

2021 - 2022

WAKIL KETUA 1

- Membantu Ketua dalam Pembuatan, pelaksanaan serta pengawasan terhadap berjalannya program kerja.

- Menjalin hubungan baik dengan organisasi lain.
- Melakukan koordinasi dan inisiasi kerja sama dengan organisasi lain untuk kemajuan organisasi.

PENGURUS ASRAMA MTS PUTRA PONDOK PESANTREN KRAPYAK YAYASAN ALI MAKSUM 2015 - 2022

KOORDINATOR BENDAHARA

- Merancang, mengelola dan mendistribusikan Anggaran tahunan untuk tiga asrama
- Melakukan koordinasi dengan seluruh divisi terkait kebutuhan anggaran dan pendistribusiannya.
- Melakukan koordinasi langsung dengan bendahara yayasan terkait dengan anggaran asrama.
- Melakukan Pelaporan rutin terhadap penggunaan anggaran kepada pengasuh pondok pesantren.

IKATAN ALUMNI MA ALI MAKSUM YOGYAKARTA ANGKATAN 2015

2018 - sekarang

KETUA

- Merancang serta merealisasikan program kerja tahunan.
- Menjaga hubungan baik dengan seluruh alumni.
- Melakukan koordinasi terkait agenda - agenda terdekat.

PERGERAKAN MAHASISWA ISLAM INDONESIA (PMII) GADJAH MADA

2018 - 2019

KOORDINATOR DIVISI KADERISASI

- Membuat dan menjalankan bersama tim program kerja yang telah disusun.
- Melakukan koordinasi dengan pengurus rayon divisi kaderisasi.
- Merancang strategi rekrutmen yang efektif dan tepat sasaran di awal tahun ajaran baru.

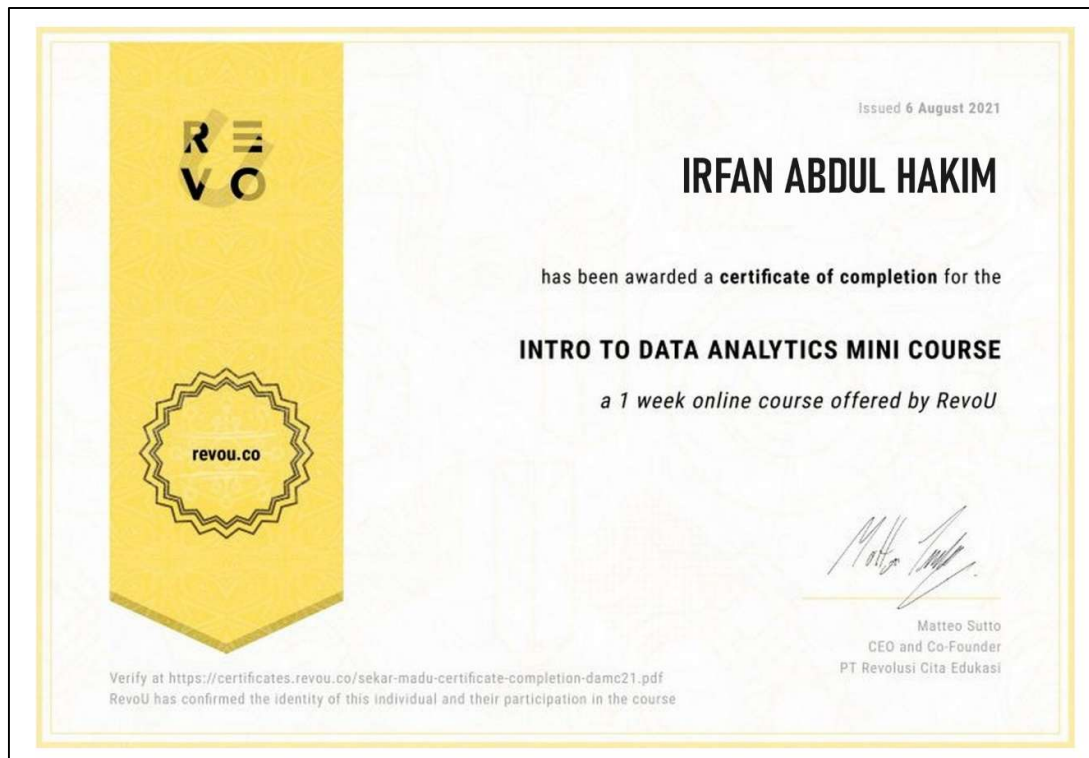
PENGALAMAN KESUKARELAAN

- **Survey Kepuasan Pelanggan PDAM Yogyakarta** - Yogyakarta (Februari 2022): Koordinator Enumerator
- **Seminar Nasional Menjadi Wirausahawan Sukses** - Yogyakarta (September 2020): Moderator
- **Penerimaan Santri Baru Yayasan Ali Maksu** - Yogyakarta (Januari - Februari 2020): Divisi Logistik
- **Riset Pusat Studi Pancasila Universitas Pembangunan Nasional Yogyakarta** - Yogyakarta (Agustus- September 2019): Enumerator
- **Pelatihan Kader Dasar PMII Gajah Mada** - (Maret 2018): Sterring Comitte
- **Pekan Raya Anak** - Yogyakarta (Desember 2017): Bendahara
- **Kulo Nuwun Party** - Yogyakarta (Maret 2016): Divisi Logistik

PENGALAMAN RISET

- Penulisan Buku Jusuf Kalla: Sang Pendamai Ulung
- Anotasi Data Sentiment Tweet [Analisis Sentiment Neural Network] dengan Kata Kunci Kopi Sianida
- Anotasi Data Sentiment Tweet [Analisis Sentiment Neural Network] dengan Kata Kunci IKN
- Anotasi Data Sentiment Tweet [Analisis Sentiment Neural Network] dengan Kata Kunci Bitcoin Halving
- Anotasi Data Sentiment Tweet [Analisis Sentiment Neural Network] dengan Kata Kunci Hak Angket
- Anotasi Data Sentiment Tweet [Analisis Sentiment Neural Network] dengan Kata Kunci Quick Count

L2 Sertifikat Expert





#DQLABPRJ8 JIWHNQ

CERTIFICATE OF COMPLETION

This certificate is proudly presented to

IRFAN ABDUL HAKIM

Has Completed in
Project Data Analysis for Finance: Performa Cabang

Sep 8, 2020

L3 Surat Ijin Penelitian



UMKKT
Program Studi
Teknik Informatika
Fakultas Sains dan Teknologi

Telp. 0541-748511 Fax. 0541-766832

Website <http://informatika.umkt.ac.id>

email: informatika@umkt.ac.id



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 055-003/KET/FST.1/A/2024

Lampiran : -

Perihal : **Keterangan Pengambilan Data Sekunder**

Assalamu 'alaikum Warrahmatullahi Wabarrakatuh

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Amin.

Dengan surat ini, kami menerangkan bahwa mahasiswa berikut:

Nama : Muhammad Rayhan Elfansyah

NIM : 2011102441075

Program Studi : Teknik Informatika

Melakukan penelitian dengan pengambilan data sekunder di Google Playstore data yang diambil yaitu data ulasan pengguna aplikasi DANA.

Demikian hal ini disampaikan, atas kerjasamanya kami ucapkan terima kasih.

Wassalamu 'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 20 Dzulhijjah 1445 H

27 Juni 2024 M

Dua Program Studi S1 Teknik Informatika



Muhammad Rayhan Elfansyah
Elfansyah, S.Kom., M.TI
IDN. 1118019203

Kampus 1 : Jl. Ir. H. Juanda, No.15, Samarinda
Kampus 2 : Jl. Pelita, Pesona Mahakam, Samarinda

L4 Code Pengumpulan Data

```
#Referensi: https://www.linkedin.com/pulse/how-scrape-google-play-reviews-4-simple-steps-using-python-kundi/
#download library google-play-scraper
!pip install google-play-scraper

Collecting google-play-scraper
  Downloading google_play_scraper-1.2.7-py3-none-any.whl (28 kB)
Installing collected packages: google-play-scraper
Successfully installed google-play-scraper-1.2.7

[ ] from google_play_scraper import app

import pandas as pd

import numpy as np

[ ] #scrape jumlah ulasan yang diinginkan
from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'id.dana',
    lang='id', #disini kita mau men scrape data ulasan aplikasi shopee yang berada di google play store
    country='id', #kita setting bahasa nya menjadi bahasa indonesia
    sort=Sort.MOST_RELEVANT, # # kemudian kita gunakan most_relevant untuk mendapatkan ulasan yang paling relevant
    count=700, # disini jumlah ulasan yang mau kita ambil ada seribu
    filter_score_with=None # # kemudian di filter_score kita gunakan None untuk mengambil semua score atau rating bintang 1 sampai 5
)
```

```
[ ] df_busu = pd.DataFrame(np.array(result), columns=['review'])

df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))

df_busu.head()
```

```
[ ] len(df_busu.index) #kemudian hitung kembali berapa jumlah data yg didapatkan

700

[ ] df_busu[['userName', 'score', 'at', 'content']].head() #preview userName, rating, date-time, and reviews only
```

```
[ ] #Run This Code to Sort the Data By Date

new_df = df_busu[['userName', 'score', 'at', 'content']]
sorted_df = new_df.sort_values(by='at', ascending=False) #Sort by Newst, change to True if you want to sort by Oldest.
sorted_df.head()
```

```
[ ] my_df = sorted_df[['userName', 'score', 'at', 'content']] #get userName, rating, date-time, and reviews only

[ ] my_df.head()
```

```
[ ] new_df.to_csv("scrapped_data.csv", index = False) #kemudian save menjadi file csv
```

L5 Install Library

```
import re # Modul untuk regular expression
import pandas as pd # Manipulasi data tabular dengan Pandas
import numpy as np # Operasi numerik dengan NumPy
import nltk # Pemrosesan teks dengan NLTK

!pip install Sastrawi # Paket Sastrawi untuk pemrosesan bahasa Indonesia

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory # Stemming bahasa Indonesia dari Sastrawi
from nltk.corpus import stopwords # Stopwords dari NLTK
from nltk.tokenize import word_tokenize # Tokenisasi teks dari NLTK

nltk.download('punkt') # Model tokenisasi teks dari NLTK
nltk.download('stopwords') # Daftar stopwords dari NLTK
```


L6 Membaca Dataset

```
[ ] data = pd.read_csv("Dana CSV.csv")
    data.head(700)
```

L7 Cek Informasi Dataset

```
▶ # Membaca dan memuat data dari file CSV "Dana CSV.csv"
data = pd.read_csv("Dana CSV.csv")
data.info() # Menampilkan informasi tentang struktur data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   User Nama    700 non-null   object
1   Waktu        700 non-null   object
2   Komentar     700 non-null   object
3   SENTIMEN     700 non-null   object
dtypes: object(4)
memory usage: 22.0+ KB
```

L8 Cek Jumlah Sentimen Positif, Negatif, Dan Netral

```
[ ] sentiment_counts = data['SENTIMEN'].value_counts() # Menghitung Jumlah Kemunculan sentimen positif dan negatif

POSITIF = sentiment_counts.get('POSITIF', 0)
NEGATIF = sentiment_counts.get('NEGATIF', 0)
NETRAL = sentiment_counts.get('NETRAL', 0)

print(f"Total Sentimen POSITIF: {POSITIF}") # mencetak jumlah total sentimen positif.
print(f"Total Sentimen NEGATIF: {NEGATIF}") # mencetak jumlah total sentimen negatif.
print(f"Total Sentimen NETRAL: {NETRAL}") # mencetak jumlah total sentimen netral.
```

L9 Membaca Dataset Kolom Review dan Sentimen

```
[ ] columns = ['Komentar', 'SENTIMEN']
df = pd.DataFrame(data, columns=columns).iloc[0:700] # membatasi DataFrame hanya pada 700 baris pertama.
df.head(5) # Menampilkan 5 baris pertama dari DataFrame df untuk melihat preview data.
```

L10 Pre-Processing Data

```
[ ] def case_folding(text):
    # Mengubah teks menjadi huruf kecil
    if isinstance(text, str):
        lowercase_text = text.lower()
        return lowercase_text
    else:
        return text

df['case_folding'] = df['Komentar'].apply(case_folding)
df.head(5)
```

```
[ ] def clean_text(tweet):
    tweet = re.sub(r'https?:\/\/\S+', '', tweet)
    tweet = re.sub(r'^a-zA-Z', '', tweet)

    emoji_pattern = re.compile("[
        u"\U0001F600-\U0001F64F" # emoticons
        u"\U0001F300-\U0001F5FF" # symbols & pictographs
        u"\U0001F680-\U0001F6FF" # transport & map symbols
        u"\U0001F1E0-\U0001F1FF" # flags (iOS)
    ]+", flags=re.UNICODE)
    tweet = emoji_pattern.sub(r'', tweet)

    tweet = re.sub(r'\d+', '', tweet)
    tweet = re.sub(r'\b\w\b', '', tweet)
    tweet = re.sub(r'\s+', '', tweet).strip()

    return tweet

df['cleansing'] = df['case_folding'].apply(clean_text)
df.head(5)
```

```
[ ] def tokenize(text):
    return word_tokenize(text)

df['tokenizing'] = df['cleansing'].apply(tokenize)
df.head(5)
```

```
[ ] stop_words = stopwords.words('indonesian')

additional_stopwords = [
    'yg', 'b', 'd', 'g', 'nya', 'ya', 'sih', 'aku', 'kamu', 'dia', 'mereka',
    'kita', 'saya', 'kalian', 'ini', 'itu', 'loh', 'dong', 'kan', 'uh', 'aa',
    'uh', 'kak', 'ah', 'oh', 'hmm', 'eh', 'aduh', 'ad', 'deh', 'hehe', 'ehh',
    'gk', 'h', 'hh', 'huh', 'udah', 'mau', 'gw', 'lu', 'lo', 'loh', 'pake', 'ntah',
    'ma', 'km', 'kalo', 'kl', 'tuh', 'set', 'dll', 'mcm', 'mn', 'gan', 'bro', 'sis', 'de',
    'btw', 'kayak', 'spt', 'bilang', 'kata', 'tau', 'lihat', 'bikin', 'dan', 'atau',
    'setelah', 'juga', 'ts', 'btul', 'tg', 'tr', 'utk', 'kmrn', 'tdk', 'sy', 'sm', 'dm',
    'hrs', 'dg', 'dl', 'dt', 'sdh', 'tdk', 'tp', 'bsa', 'ga', 'gak', 'masuk',
    'aja', 'apk', 'buka'
]

stop_words.extend(additional_stopwords)

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

df['stopword_removal'] = df['tokenizing'].apply(lambda x: remove_stopwords(x))
df.head(5)
```

```
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(tokens):
    return [stemmer.stem(word) for word in tokens]

df['stemming'] = df['stopword_removal'].apply(stem_text)
df['stemming'] = df['stemming'].apply(lambda x: ' '.join(x))
df.head(5)
```


L11 Pembobotan Kata (TF-IDF)

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

    # Memuat data yang telah diproses sebelumnya
    documents = df['stemming'].tolist()
    N = len(documents)

    # Inisialisasi TF-IDF Vectorizer
    vectorizer = TfidfVectorizer()

    # Melakukan fit dan transformasi pada dokumen
    tfidf_matrix = vectorizer.fit_transform(documents)

    # Mendapatkan nama fitur (terms)
    terms = vectorizer.get_feature_names_out()

    # Mendapatkan nilai IDF
    idf_values = vectorizer.idf_

    # Mengonversi matriks TF-IDF menjadi DataFrame
    tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=terms)

    # Menggabungkan DataFrame TF-IDF dengan label sentimen
    final_df = pd.concat([tfidf_df, df['SENTIMEN']], axis=1)

    # Mencetak 700 baris pertama dari DataFrame akhir
    print(final_df.head(700))
```

```
▶ # Ambil daftar kata kunci (terms)
    terms = vectorizer.get_feature_names_out()

    # Menghitung frekuensi setiap istilah dalam seluruh dataset
    term_frequencies = tfidf_matrix.sum(axis=0).A1
    terms_frequencies_df = pd.DataFrame({'term': terms, 'frequency': term_frequencies})

    # Mengurutkan istilah berdasarkan frekuensi dalam urutan menurun
    sorted_terms = terms_frequencies_df.sort_values(by='frequency', ascending=False)

    # Menampilkan istilah yang paling sering muncul
    print("Most frequent terms:")
    print(sorted_terms.head(10))
```

L12 Hasil Akurasi Nilai K

```
▶ from sklearn.neighbors import KNeighborsClassifier
    from sklearn.metrics import accuracy_score

    # Mencari akurasi untuk berbagai nilai k
    k_values = range(1, 21)
    accuracies = []

    for k in k_values:
        knn_classifier = KNeighborsClassifier(n_neighbors=k)
        knn_classifier.fit(X_train, y_train)
        y_pred = knn_classifier.predict(X_test)
        accuracy = accuracy_score(y_test, y_pred)
        accuracies.append((k, accuracy))

    # Print hasil akurasi untuk setiap nilai k
    print("Akurasi untuk berbagai nilai k:")
    for k, accuracy in accuracies:
        print(f"Nilai K: {k}, Akurasi: {accuracy:.4f}")
```

L13 Grafik Akurasi Nilai K

```
[ ] # Mencari akurasi untuk berbagai nilai k
k_values = range(1, 21)
accuracies = []

for k in k_values:
    knn_classifier = KNeighborsClassifier(n_neighbors=k)
    knn_classifier.fit(X_train, y_train)
    y_pred = knn_classifier.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    accuracies.append(accuracy)

# Plotting akurasi terhadap nilai k
plt.figure(figsize=(10, 6))
plt.plot(k_values, accuracies, marker='o')
plt.title('Akurasi terhadap Nilai K')
plt.xlabel('Nilai K')
plt.ylabel('Akurasi')
plt.xticks(k_values)
plt.grid(True)
plt.show()
```

L14 Split Data dan Evaluasi Confusion Matrix Metode KNN

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

# Membagi data menjadi set pelatihan dan pengujian (70% pelatihan, 30% pengujian)
X_train, X_test, y_train, y_test = train_test_split(final_df.drop('SENTIMEN', axis=1),
                                                    final_df['SENTIMEN'],
                                                    test_size=0.3,
                                                    random_state=42)

# Menginisialisasi classifier KNN dengan parameter yang disesuaikan
knn_classifier = KNeighborsClassifier(n_neighbors=5) # Sesuaikan jumlah neighbors sesuai kebutuhan

# Melatih classifier
knn_classifier.fit(X_train, y_train)

# Prediksi pada set pengujian
y_pred = knn_classifier.predict(X_test)

# Akurasi
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Menghitung dan memvisualisasikan confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['POSITIF', 'NEGATIF', 'NETRAL'], yticklabels=['POSITIF', 'NEGATIF', 'NETRAL'])
plt.xlabel('Prediksi')
plt.ylabel('Actual')
plt.title('Confusion Matrix (KNN)')
plt.show()
```

L15 Split Data dan Evaluasi Confusion Matrix Metode Naive Bayes

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB # Import Gaussian Naive Bayes
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

# Membagi data menjadi set pelatihan dan pengujian (70% pelatihan, 30% pengujian)
X_train, X_test, y_train, y_test = train_test_split(final_df.drop('SENTIMEN', axis=1),
                                                  final_df['SENTIMEN'],
                                                  test_size=0.3,
                                                  random_state=42)

# Menginisialisasi classifier Naive Bayes (Gaussian)
nb_classifier = GaussianNB()

# Melatih classifier
nb_classifier.fit(X_train, y_train)

# Prediksi pada set pengujian
y_pred = nb_classifier.predict(X_test)

# Akurasi
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Menghitung dan memvisualisasikan confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['POSITIF', 'NEGATIF', 'NETRAL'], yticklabels=['POSITIF', 'NEGATIF', 'NETRAL'])
plt.xlabel('Prediksi')
plt.ylabel('Actual')
plt.title('Confusion Matrix (Naive Bayes)')
plt.show()
```

L16 Visualisasi Wordcloud Positif, Negatif, Dan Netral

```
[ ] import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Menginisialisasi TF-IDF Vectorizer
vectorizer = TfidfVectorizer()

# Menyesuaikan dan mengubah dokumen
tfidf_matrix = vectorizer.fit_transform(documents)

# Mendapatkan nama fitur (istilah)
terms = vectorizer.get_feature_names_out()

# Membuat DataFrame dari matriks TF-IDF
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=terms)

# Menggabungkan DataFrame TF-IDF dengan label sentimen
final_df = pd.concat([tfidf_df, df['SENTIMEN']], axis=1)

# Membuat DataFrame terpisah untuk setiap kategori sentimen
sentiments = df['SENTIMEN'].unique() # Mendapatkan kategori sentimen yang unik

# Membuat fungsi untuk menghasilkan dan menampilkan WordCloud
def generate_wordcloud(text, title):
    wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(text)
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(title)
    plt.show()

# Menghasilkan WordCloud untuk setiap kategori sentimen
for sentiment in sentiments:
    # Memilih baris berdasarkan sentimen
    subset_df = final_df[final_df['SENTIMEN'] == sentiment]







    # Menghitung frekuensi kata (jumlah nilai TF-IDF per kolom)
    word_freq = subset_df.drop(columns=['SENTIMEN']).sum(axis=0).to_dict()




    # Menghasilkan dan menampilkan WordCloud
    generate_wordcloud(word_freq, f"WordCloud - SENTIMEN {sentiment}")
```

L17 Kartu Kendali Bimbingan

KARTU KENDALI BIMBINGAN

Nama Mahasiswa : Muhammad Rayhan Elfansyah
 NIM : 2011102441075
 Nama Dosen Pembimbing : Rudiman, S.Kom, M.Sc
 Judul Penelitian : Perbandingan Metode K-Nearest Neighbor (KNN) dan Naïve Bayes Terhadap Analisis Sentiment Pada Pengguna Aplikasi DANA Menggunakan Fitur Ekstraksi TF-IDF

NO	TANGGAL	URAIAN PEMBIMBINGAN	PARAF DOSEN
1	7 Feb 2024	Bimbingan pertama mendiskusikan persetujuan bimbingan dengan dosen Bapak Rudiman S.Kom, M.Sc. dengan topik penelitian Data Sentiment Analyst	
2	14 Feb 2024	Mencari permasalahan topik yang akan diteliti untuk dijadikan judul	
3	22 Feb 2024	Melakukan tahap pencarian data menggunakan pyhton di Google Colab	
4	29 Feb 2024	Menentukan judul dan mulai perencanaan latar belakang dengan syarat ketentuan yang telah ditetapkan oleh dosen pembimbing, yaitu, menggunakan sumber jurnal paling sedikit 13 dan maksimal 15 jurnal untuk BAB 1 dan bersinta minimal sinta 4	
5	9 Maret 2024	Dalam pengerjaan latar belakang harus mengikuti aturan yang telah ditetapkan oleh dospem	
6	13 Maret 2024	Melakukan beberapa revisi terhadap mahasiswa yang dibimbing	
7	18 Maret 2024	Memberikan beberapa saran terhadap canvas pengajuan judul ke prodi	
8	27 Maret 2024	Memberikan beberapa revisi terhadap latar belakang rumusan masalah batasan masalah tujuan penelitian dan manfaat penelitian	

9	5 April 2024	Masuk BAB II menggunakan strategi yang telah diberikan oleh dosen pembimbing menggunakan pembobotan TF-IDF dalam penelitian yang dilakukan	
10	9 April 2024	Memberi masukan dan revisi dan bab II	
11	24 April 2024	Membahas target penelitian dan mencari sumber jurnal serta memberikan revisian pada BAB I dan BAB II	

Dosen Pembimbing


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

Mengetahui,
 Ketua Program Studi



Arbansyah, S.Kom, M.TI
 NIDN. 118019203

SKRIPSI MUHAMMAD RAYHAN ELFANSYAH

by Teknik Informatika Universitas Muhammadiyah Kalimantan Timur



Submission date: 25-Jul-2024 09:25AM (UTC+0800)

Submission ID: 2422037828

File name: SKRIPSI_MUHAMMAD_RAYHAN_ELFANSYAH.docx (900.82K)

Word count: 8125

Character count: 50749

SKRIPSI MUHAMMAD RAYHAN ELFANSYAH



ORIGINALITY REPORT

22%

SIMILARITY INDEX

19%

INTERNET SOURCES

13%

PUBLICATIONS

8%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Brawijaya Student Paper	2%
2	ejournal.unesa.ac.id Internet Source	1%
3	repository.ub.ac.id Internet Source	1%
4	openjournal.unpam.ac.id Internet Source	1%
5	repository.uin-suska.ac.id Internet Source	1%
6	docplayer.info Internet Source	1%
7	123dok.com Internet Source	1%
8	Submitted to STMIK STIKOM Bali Student Paper	1%
9	doaj.org Internet Source	1%

RIWAYAT HIDUP



Muhammad Rayhan Elfansyah, Lahir di Samarinda pada Tanggal 07 Juli 2002. Merupakan anak pertama dari tiga bersaudara yang lahir dari pasangan Bapak Fadliansyah dan Ibu Fatmawati. memiliki saudara bernama Muhammad Naufal Rozansyah dan Muhammad Dzikri Romansyah. Pendidikan yang telah di tempuh oleh peneliti yaitu SD Negeri 028 Samarinda lulus pada tahun 2014. Kemudian dilanjutkan dengan menempuh Pendidikan di MTS DDI Tani Aman lulus pada tahun 2017. Setelah itu dilanjutkan dengan menempuh Pendidikan di SMA Negeri 7 Samarinda, lulus pada tahun 2020. Pada tahun 2020 peneliti menempuh Pendidikan Program Studi S1 Teknik Informatika di Universitas Muhammadiyah Kalimantan Timur sebagai syarat kelulusan dan memperoleh gelar Sarjana Komputer (S.Kom).