

LAMPIRAN

Lampiran 1 CV *Expert Labeling*

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

Lampiran 2 Surat Izin Penelitian

 UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR Berkeadilan Berprestasi Berkeagamaan	UMKT 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 : 055-005/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 : Ikhsan Nuttakwa Takbirata Ihram Nabawi
NIM : 2011102441072
Program Studi : Teknik Informatika


Melakukan penelitian dengan pengambilan data sekunder di Google Maps data yang diambil yaitu data ulasan pelayanan BPJS.


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

Wassalamu'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 20 Dzulhijjah 1445 H
27 Juni 2024 M

Ketua Program Studi S1 Teknik Informatika


Arhansyah, S.Kom., M.TI
IDN. 1118019203



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

CS Dipindai dengan CamScanner

Lampiran 3 Pengumpulan Data

```
[1] feature_id = "0x2df67f574f743fe1:0x9d1617309fd1d93a"
    loop = 50

pip install beautifulsoup4
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (4.12.3)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4) (2.5)

[3] from datetime import datetime, timedelta

# nama hari dalam bahasa Indonesia
nama_hari_indonesia = {
    "Monday": "Senin",
    "Tuesday": "Selasa",
    "Wednesday": "Rabu",
    "Thursday": "Kamis",
    "Friday": "Jumat",
    "Saturday": "Sabtu",
    "Sunday": "Minggu"
}
```

```
# Fungsi untuk mengonversi teks waktu menjadi timedelta
def konversi_teks_ke_timedelta(teks):
    if 'hari' in teks:
        jumlah = int(teks.split()[0])
        return timedelta(days=jumlah)
    elif 'minggu' in teks:
        if 'seminggu' in teks:
            return timedelta(weeks=1)
        else:
            jumlah = int(teks.split()[0])
            return timedelta(weeks=jumlah)
    elif 'bulan' in teks:
        if 'sebulan' in teks:
            return timedelta(days=30)
        else:
            jumlah = int(teks.split()[0])
            return timedelta(days=jumlah*30)
    else:
        raise ValueError("Format teks waktu tidak dikenali")

# Fungsi untuk mengonversi teks waktu menjadi tanggal dan hari
def konversi_teks_ke_tanggal(teks):
    if 'hari' in teks or 'minggu' in teks or 'bulan' in teks:
        timedelta_lalu = konversi_teks_ke_timedelta(teks)
        tanggal = datetime.now() - timedelta_lalu
        nama_hari = tanggal.strftime("%A")
        nama_hari_indo = nama_hari_indonesia.get(nama_hari, nama_hari)
        return tanggal, nama_hari_indo
    elif 'tahun' in teks:
        if 'setahun' in teks:
```

```

        return datetime.now() - timedelta(days=365), None
    else:
        jumlah_tahun = int(teks.split()[0])
        return datetime.now() - timedelta(days=365 * jumlah_tahun), None
    else:
        # Jika format tidak sesuai dengan yang dikenali, kembalikan None
        return None, None

import requests
import re
from datetime import datetime, timedelta
from bs4 import BeautifulSoup
import pandas as pd

def get_reviews_data(feature_id):
    token = ''
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 Safari/537.36"
    }

    base_url = "https://www.google.com/async/reviewDialog?hl=id&async=feature_id:{feature_id},next_page_token:{token},sort_by:newest,start_index:,associated_topic:,fmt:pc"

    # List untuk menyimpan DataFrame dari setiap permintaan
    dfs = []

```

```

# Lakukan loop permintaan
for i in range(loop):
    response = requests.get(base_url.format(feature_id=feature_id, token=token), headers=headers)
    soup = BeautifulSoup(response.content, 'html.parser')

    user = []
    location_info = {}
    data_id = ''

    for el in soup.select('.c9QyIf'):
        data_id = soup.select_one('.DvzRrc')['data-fid']
        next_page_token = soup.select_one('.gws-localreviews__general-reviews-block')['data-next-page-token']
        if next_page_token:
            token = next_page_token
        else:
            token = None # Jika token berikutnya kosong, hentikan loop
            break

    for el in soup.select('.gws-localreviews__google-review'):
        # try:
        rating_label = el.select_one('span.lTi8oc.z3Hnkc')['aria-label']
        match = re.search(r'(\d+)(?:,\d+)?', rating_label)

```

```

if match:
    rating = match.group(1)
else:
    rating = "0"
# except KeyError:
#     rating = "0"

waktu = el.select_one('.dehysf').text.strip()
# print(waktu)
tanggal, nama_hari_indo = konversi_teks_ke_tanggal(waktu)
# print(tanggal)
# if tanggal is None: # Tambahkan penanganan eksepsi di sini
#     continue # Lewati data yang tidak valid

tanggal_formatted = tanggal.strftime("%d-%m-%Y")
time_formatted = f"{nama_hari_indo} {tanggal_formatted}"
sentimen = 'Negatif' if int(rating) <= 3 else 'Positif'
user.append({
    'name': el.select_one('.TSuDb').text.strip(),
    'rating': rating,
    'review': el.select_one('.Jtu6Td').text.strip(),
    'time': time_formatted,
    'sentimen': sentimen
})

```

```

df = pd.DataFrame(user)

# Tambahkan DataFrame ke dalam list
dfs.append(df)
total_data += len(df)

print(f"Loop ke-{i+1}:")
# print(df)
print("-----")

if token is None:
    break # Hentikan loop jika token berikutnya kosong

# Gabungkan semua DataFrame menjadi satu
combined_df = pd.concat(dfs, ignore_index=True)

# Simpan DataFrame gabungan ke dalam file Excel
combined_df.to_excel("review_data.xlsx", index=False)

print("Total data yang sudah didapat:", total_data)
print("Data telah disimpan dalam review_data.xlsx")

get_reviews_data(feature_id)

```

Lampiran 4 *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

Requirement already satisfied: Sastrawi in /usr/local/lib/python3.10/dist-packages (1.0.1)
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
True

```

Lampiran 5 Membaca Dataset

```

[41] df = pd.read_csv("Review_Data_Baru.csv", encoding='latin1')
df.head(500)

```

Lampiran 6 Cek Informasi Dataset

```
[42] import pandas as pd

# Mencoba membaca file CSV dengan encoding 'latin1'
data = pd.read_csv("Review_Data_Baru.csv", encoding='latin1')
data.info() # Menampilkan informasi tentang struktur data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ---
 0   name        500 non-null    object
 1   rating      500 non-null    int64
 2   review      500 non-null    object
 3   time        500 non-null    object
 4   sentimen    500 non-null    object
dtypes: int64(1), object(4)
memory usage: 19.7+ KB
```

Lampiran 7 Cek Jumlah Sentimen Positif dan Negatif

```
[43] 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)

print(f"Total Sentimen Positif: {positif}") # mencetak jumlah total sentimen positif.
print(f"Total Sentimen Negatif: {negatif}") # mencetak jumlah total sentimen negatif.
```

```
Total Sentimen Positif: 487
Total Sentimen Negatif: 13
```

Lampiran 8 Persentase Sentimen

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Misalkan dataset Anda bernama 'Review_Data_Baru' dan kolom targetnya bernama 'sentimen'
df = pd.read_csv('Review_Data_Baru.csv', encoding='latin1')

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

# Membuat pie chart untuk persentase sentimen
plt.figure(figsize=(10, 8))
colors = sns.color_palette('viridis', len(sentimen_counts))
sentimen_counts.plot(kind='pie', autopct='%1.1f%%', startangle=140, colors=colors, labels=['Positif', 'Negatif'], wedgeprops={'edgecolor': 'black'})

plt.title('Persentase Sentimen', fontsize=16)
plt.ylabel('') # Menghilangkan label default pada y-axis
plt.legend(title='sentimen', loc='upper left')
plt.show()
```

Lampiran 9 Membaca Dataset Kolom Review dan Sentimen

```
[45] columns = ['review', 'sentimen']
df = pd.DataFrame(data, columns=columns).iloc[0:500] # membatasi DataFrame hanya pada 500 baris pertama.
df.head(500) # Menampilkan 5 baris pertama dari DataFrame df untuk melihat preview data.
```

Lampiran 10 Pre-Processing Data

```
PRE-PROCESSING DATA

CASE FOLDING

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['review'].apply(case_folding)
df.head(5)

Tampilkan output tersembunyi

Langkah berikutnya:  

CLEANING

[47] def clean_text(tweet):
    tweet = re.sub(r'https?:\V\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)
```

```
TOKENISASI

def tokenize(text):
    return word_tokenize(text)

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

Tampilkan output tersembunyi

Langkah berikutnya:  

STOPWORD

[49] import pandas as pd
from nltk.corpus import stopwords
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# Assuming the dataframe 'df' and column 'token' already exist

# Define stopwords
stop_words = stopwords.words('indonesian')

additional_stopwords = [
    'yg','bi','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','mm','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','muas',
]

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)
```


STEMMING

```
[50] from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

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

def stem_text(tokens):
    # Mengembalikan kata dasar untuk setiap token
    return [stemmer.stem(word) for word in tokens]

# Terapkan fungsi stem_text ke kolom 'stopword_removal' dan gabungkan kembali menjadi string
df['stemming'] = df['stopword_removal'].apply(stem_text)
df['stemming'] = df['stemming'].apply(lambda x: ' '.join(x))

# Tampilkan 5 baris pertama dari DataFrame
df.head(5)
```

Lampiran 11 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 500 baris pertama dari DataFrame akhir
print(final_df.head(500))

Tampilkan output tersembunyi

AMPILKAN KATA-KATA YANG SERING MUNCUL

# 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))
```

Lampiran 12 Split Data dan Evaluasi Confusion Matrix Metode KNN

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Membagi data menjadi set pelatihan dan pengujian (80% pelatihan, 20% 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'], yticklabels=['POSITIF', 'NEGATIF'])
plt.xlabel('Prediksi')
plt.ylabel('Actual')
plt.title('Confusion Matrix (KNN)')
plt.show()
```

Lampiran 13 Hasil Akurasi Pada 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}")
```

Lampiran 14 Plot 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()
```

Lampiran 15 Visualisasi *Wordcloud* Positif dan Negatif

```
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}")
```

Lampiran 16 Kartu Kendali Bimbingan

KARTU KENDALI BIMBINGAN

Nama Mahasiswa : Ikhsan Nuttakwa Takbirata Ihram Nabawi
NIM : 2011102441072
Nama Dosen Pembimbing : Rudiman, S.Kom, M.Sc
Judul Penelitian : Analisis Sentimen pada Ulasan Aplikasi Google Maps terhadap Pelayanan Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan Samarinda Menggunakan Metode K-Nearest Neighbor dengan 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 python 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



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

Mengetahui,
Ketua Program Studi



Arbansyah, S.Kom., M.TI
NIDN. 1118019203

SKRIPSI IKHSAN NUTTAKWA TAKBIRATA IHRAM NABAWI

by S1 Teknik Informatika Universitas Muhammadiyah Kalimantan Timur



Submission date: 25-Jul-2024 09:24AM (UTC+0800)

Submission ID: 2422037353

File name: SKRIPSI_IKHSAN_NUTTAKWA_TAKBIRATA_IHRAM_NABAWI.docx (1.03M)

Word count: 5312

Character count: 33815

SKRIPSI IKHSAN NUTTAKWA TAKBIRATA IHRAM NABAWI

Ail

ORIGINALITY REPORT

29%
SIMILARITY INDEX

26%
INTERNET SOURCES

14%
PUBLICATIONS

12%
STUDENT PAPERS

PRIMARY SOURCES

1	repository.dinamika.ac.id Internet Source	2%
2	repository.uin-suska.ac.id Internet Source	1%
3	ejournal.unesa.ac.id Internet Source	1%
4	jurnal.polsri.ac.id Internet Source	1%
5	docplayer.info Internet Source	1%
6	www.researchgate.net Internet Source	1%
7	123dok.com Internet Source	1%
8	jurnal.kaputama.ac.id Internet Source	1%
9	doaj.org Internet Source	1%

RIWAYAT HIDUP



Ikhsan Nuttakwa Takbirata Ihram Nabawi, lahir di samarinda pada tanggal 09 November 2001. Merupakan anak kedua dari dua bersaudara yang lahir dari pasangan Bapak Masrul dan Ibu Erna M. memiliki saudari bernama Dina Mi'ratul Jannata Rayyana Nabawiyah. Pendidikan yang telah di tempuh oleh peneliti yaitu MI Sullamul Ulum Samarinda lulus pada tahun 2014. Kemudian dilanjutkan dengan menempuh Pendidikan di SMP Negeri 16 Samarinda lulus pada tahun 2017. Setelah itu dilanjutkan dengan menempuh Pendidikan di SMA Negeri 8 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).