

## Daftar Lampiran

### L1 Pengumpulan Data

```
▶ | feature_id = "0x2df67ace57e958ef:0xa9fb2adcb21c8264"  
   | loop = 60
```

```
[ ] pip install beautifulsoup4
```

```
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_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 = []

    total_data = 0

    # 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('.DvzRnc')['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}"
# Determine sentiment
if int(rating) <= 3:
    sentimen = 'Negatif'
elif int(rating) == 4:
    sentimen = 'Netral'
else:
    sentimen = 'Positif'
user.append({
    'name': el.select_one('.TSubDb').text.strip(),
    'rating': rating,
    'review': el.select_one('.Jtu6Td').text.strip(),
    'time': time_formatted,
    'sentimen': sentimen
})

df = pd.DataFrame(user)

```

```

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("googlemaps_data.xlsx", index=False)

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

get_reviews_data(feature_id)

```

## L2 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

```

### L3 Cek Informasi Dataset

```
import pandas as pd

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

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 502 entries, 0 to 501
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   name        502 non-null   object
1   rating      502 non-null   int64
2   review      502 non-null   object
3   time        502 non-null   object
4   sentimen    502 non-null   object
dtypes: int64(1), object(4)
memory usage: 19.7+ KB
```

### L4 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)
netral = sentiment_counts.get('Netral', 0)
negatif = sentiment_counts.get('Negatif', 0)

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

### L5 Membaca Dataset Kolom Review dan Sentimen

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

### L6 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['review'].apply(case_folding)
df.head(5)
```

```

def clean_text(review):
    # Remove URLs
    review = re.sub(r'https?:\//\S+', '', review)
    # Remove non-alphabetic characters
    review = re.sub(r'^a-zA-Z', ' ', review)

    # Remove emojis
    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)
    review = emoji_pattern.sub(r'', review)

    # Remove digits
    review = re.sub(r'\d+', '', review)
    # Remove single characters
    review = re.sub(r'\b\w\b', '', review)
    # Remove extra spaces
    review = re.sub(r'\s+', ' ', review).strip()

    return review

# Applying the function to the 'case_folding' column
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', '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', 'sy', 'sm', 'dm',
    'hrs', 'dg', 'dl', 'dt', 'sdh', 'tp', 'bsa', 'dan', 'yang'
]

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)

```

## L7 Visualisasi Wordcloud

```

import pandas as pd
import numpy as np
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt

# Read CSV data
df = pd.read_csv('data.csv')

# Rename the column 'label' to 'sentimen'
df.rename(columns={'label': 'sentimen'}, inplace=True)

# Handle missing values
df['stemming'] = df['stemming'].fillna('').astype(str)

# Combine text based on sentimen
positive_text = ' '.join(df[df['sentimen'] == 'Positif']['stemming'].tolist())
neutral_text = ' '.join(df[df['sentimen'] == 'Netral']['stemming'].tolist())
negative_text = ' '.join(df[df['sentimen'] == 'Negatif']['stemming'].tolist())

# Define stopwords
stopwords = set(STOPWORDS)
stopwords.update(['https', 'co', 'RT', '...', 'amp'])

# Create WordCloud function
def create_wordcloud(text, title):
    wc = WordCloud(stopwords=stopwords, background_color="white", max_words=500, width=800, height=400)
    wc.generate(text)

    plt.figure(figsize=(10, 5))
    plt.imshow(wc, interpolation='bilinear')
    plt.axis("off")
    plt.title(title)
    plt.show()

# Generate and display WordClouds
create_wordcloud(positive_text, 'WordCloud - Positif')
create_wordcloud(neutral_text, 'WordCloud - Netral')
create_wordcloud(negative_text, 'WordCloud - Negatif')

```

## L8 Pembobotan Kata (TF-IDF)

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

# Assuming 'documents' is already defined and contains the list of stemmed texts

# Create and fit TF-IDF vectorizer
vectorizer = TfidfVectorizer()
features = vectorizer.fit_transform(documents)

# Convert the TF-IDF matrix to a DataFrame
tfidf_df = pd.DataFrame(features.toarray(), columns=vectorizer.get_feature_names_out())

# Calculate the total TF-IDF score for each document
tfidf_totals = tfidf_df.sum(axis=1)

# Add the total TF-IDF scores to the original DataFrame
df_preprocessed['total_tfidf'] = tfidf_totals

# Display the DataFrame with total TF-IDF scores
print(df_preprocessed[['stemming', 'total_tfidf']])
```

## L9 Split Data dan Evaluasi Confusion Matrix Metode Naïve Bayes

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix, 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=0.2,
                                                  random_state=42)

# Menampilkan jumlah data yang dilatih dan diuji
print(f"Jumlah data pelatihan: {len(X_train)}")
print(f"Jumlah data pengujian: {len(X_test)}")

# Visualisasi distribusi kelas pada data pelatihan
plt.figure(figsize=(14, 6))
plt.subplot(1, 2, 1)
sns.countplot(x=y_train)
plt.title('Distribusi Kelas Data Pelatihan')
plt.xlabel('Sentimen')
plt.ylabel('Frekuensi')

# Visualisasi distribusi kelas pada data pengujian
plt.subplot(1, 2, 2)
sns.countplot(x=y_test)
plt.title('Distribusi Kelas Data Pengujian')
plt.xlabel('Sentimen')
plt.ylabel('Frekuensi')

plt.show()

# Menginisialisasi classifier Naive Bayes
nb_classifier = MultinomialNB()

# Melatih classifier dengan data pelatihan
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)

# Laporan Klasifikasi
print("\nClassification Report:")
print(classification_report(y_test, y_pred, zero_division=1))

# 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=['Negatif', 'Netral', 'Positif'], yticklabels=['Negatif', 'Netral', 'Positif'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



## L10 Surat Ijin Penelitian

 <p><b>UMKT</b> Program Studi <b>Teknik Informatika</b> Fakultas Sains dan Teknologi</p>	Telp. 0541-748511 Fax.0541-766832 Website <a href="http://informatika.umkt.ac.id">http://informatika.umkt.ac.id</a> email: <a href="mailto:informatika@umkt.ac.id">informatika@umkt.ac.id</a>
---	---

*بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ*

Nomor : 055-011/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 Reifin Perdana  
NIM : 2011102441136  
Program Studi : Teknik Informatika

Melakukan penelitian dengan pengambilan data sekunder di Google Maps data yang diambil yaitu ulasan terkait jembatan repo-repo.

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

*Wassalamu'alaikum Warrahmatullahi Wabarrakatuh*

Samarinda, 21 Dzulhijjah 1445 H  
28 Juni 2024 M

Ketua Program Studi S1 Teknik Informatika

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














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

## L11 Kartu Bimbingan

### KARTU KENDALI BIMBINGAN

Nama Mahasiswa : Muhammad Reifin Perdana  
 NIM : 2011102441136  
 Nama Dosen Pembimbing : Rudiman, S.Kom, M.Sc  
 Judul Penelitian : ANALISIS SENTIMEN ULASAN JEMBATAN  
 REPO-REPO di GOOGLE MAPS  
 MENGGUNAKAN METODE NAÏVE BAYES  
 DENGAN FITUR EKSTRASI 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. 1118019203

## RIWAYAT HIDUP



Muhammad Reifin Perdana, lahir di Tenggarong pada tanggal 17 Juli 2002. Merupakan anak yang lahir dari pasangan Bapak Aripin dan Ibu Yusparini. Pendidikan yang telah di tempuh oleh peneliti yaitu SD Negeri 003 Tenggarong lulus pada tahun 2014. Kemudian dilanjutkan dengan menempuh Pendidikan di SMP Negeri 1 Tenggarong lulus pada tahun 2017. Setelah itu dilanjutkan dengan menempuh Pendidikan di SMA Negeri 1 Tenggarong, 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).