

# LAMPIRAN

## Lampiran 1 CV *Expert Labeling*

### Rizka Audina Sinaga

Jawa Tengah | [rizkasinaga17@gmail.com](mailto:rizkasinaga17@gmail.com)



#### PERSONAL INFORMASI

Berpengalaman lebih dari 4 tahun dalam keterampilan monitoring serta analisis data media online dan sosial. Selain itu, saya telah memberikan rekomendasi yang berharga untuk pengambilan keputusan terkait strategi komunikasi dan kebijakan publik kepada klien. Keahlian saya tidak hanya terbatas pada aspek analitis, tetapi juga pada kemampuan menulis yang telah terbukti melalui kontribusi saya di berbagai media, baik online, sosial, maupun cetak.

#### PENDIDIKAN

Institut Agama Islam Negeri Surakarta (2015-2019)

Komunikasi dan Penyiaran Islam

#### KEMAMPUAN

- Software Dasar Microsoft Office
- Analisis Media
- Strategi Komunikasi
- Penulisan Berita dan Konten Kreatif
- Search Engine Optimization (SEO) dan Content Management System (CMS)

#### PENGALAMAN

PT Indonesia Indicator

*Media Analyst (2020-2024)*

- Mengumpulkan, menganalisis, dan menginterpretasi data dari berbagai sumber media untuk mengidentifikasi tren dan pola yang relevan.
- Menggunakan alat analisis data dan teknologi big data untuk mengolah informasi dan menghasilkan laporan yang komprehensif.
- Memantau dan mengevaluasi sentimen publik dan tren media sosial untuk memberikan wawasan strategis kepada perusahaan.
- Menyusun dan menyampaikan laporan analisis media secara berkala kepada manajemen dan pemangku kepentingan lainnya.
- Berkolaborasi dengan tim lintas fungsi untuk mengembangkan strategi komunikasi yang efektif berdasarkan hasil analisis data.
- Mengembangkan dan mengoptimalkan metode analisis media untuk meningkatkan akurasi dan efisiensi pemantauan.
- Mengidentifikasi peluang dan ancaman yang muncul dari data media dan memberikan rekomendasi yang berdasar untuk tindakan lanjut.
- Menjaga pengetahuan terkini tentang tren teknologi big data dan analisis media untuk memastikan metodologi dan alat yang digunakan selalu mutakhir.

## Lampiran 2 Code Pengumpulan Data

```
✓ TOKEN AUTENTIKASI TWITTER
[ ] # @title TOKEN AUTENTIKASI TWITTER
    twitter_auth_token = 'd5753d38fc5d1baa12358939e1034e0d0d6f10b4'

✓ INSTALASI PANDAS DAN NODE.JS
[ ] # @title INSTALASI PANDAS DAN NODE.JS
    pip install pandas # Menginstal library pandas
    sudo apt-get update # Memperbarui daftar paket
    sudo apt-get install -y ca-certificates curl gnupg # Menginstal sertifikat (CA, curl), dan gnupg
    sudo mkdir -p /etc/apt/keyrings # Membuat direktori untuk menyimpan kunci GPG
    curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg # Mengunduh dan menyimpan kunci GPG Node
    # Menambahkan repositori NodeSource ke sources.list
    INODE_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
    sudo apt-get update # Memperbarui daftar paket lagi
    sudo apt-get install nodejs -y # Menginstal Node.js
    node -v # Menampilkan versi Node.js yang terinstal

✓ MENGUMPULKAN DATA TWEET MENGGUNAKAN TWEET-HARVEST
[ ] # @title MENGUMPULKAN DATA TWEET MENGGUNAKAN TWEET-HARVEST
    filename = 'Quick-Count.csv' # Nama file untuk menyimpan hasil pengumpulan data
    search_keyword = 'quick count lang:id until:2024-02-15 since:2024-02-14' # Kata kunci pencarian dan filter waktu
    limit = 2100 # Batas jumlah tweet yang akan dikumpulkan
    # Menjalankan tweet-harvest untuk mengumpulkan tweet
    !npx -y tweet-harvest@2.6.0 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}

✓ MEMBACA DAN MENAMPILKAN DATA TWEET
[ ] # @title MEMBACA DAN MENAMPILKAN DATA TWEET
    import pandas as pd # Mengimpor library pandas untuk manipulasi data
    file_path = f"tweets-data/{filename}" # Menentukan path file CSV
    df = pd.read_csv(file_path, delimiter=",") # Membaca file CSV ke dalam DataFrame pandas
    display(df) # Menampilkan DataFrame

✓ MENGHITUNG JUMLAH TWEET DALAM DATAFRAME
[ ] # @title MENGHITUNG JUMLAH TWEET DALAM DATAFRAME
    num_tweets = len(df) # Menghitung jumlah total tweet dalam DataFrame
    print(f"Jumlah tweet dalam dataframe adalah {num_tweets}.") # Mencetak jumlah total tweet
```

## Lampiran 3 Code *Pre-Processing*

```
LIBRARY
[] # @title LIBRARY
import re # Modul untuk regular expression
import pandas as pd # Manipulasi data tabular dengan Pandas
import numpy as np # Operasi numerik dengan NumPy

# Instalasi Sastrawi
!pip install Sastrawi # Paket Sastrawi untuk pemrosesan bahasa Indonesia
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory # Stemming bahasa Indonesia dari Sastrawi

import nltk # Pemrosesan teks dengan NLTK
from nltk.corpus import stopwords # Daftar stopwords dari NLTK
# Download data NLTK yang diperlukan
nltk.download('stopwords') # Daftar stopwords dari NLTK

MEMBACA DAN MENAMPILKAN INFORMASI DATA
[] # @title MEMBACA DAN MENAMPILKAN INFORMASI DATA
data = pd.read_csv('DATA-CSV.csv') # Membaca dan memuat data dari file CSV "Data-Quick Count.csv"
data.info() # Menampilkan informasi tentang struktur data

VISUALISASI DISTRIBUSI SENTIMEN
[] # @title VISUALISASI DISTRIBUSI SENTIMEN
import matplotlib.pyplot as plt # Menginstall library matplotlib

sentiment_counts = data['Sentimen'].value_counts() # Menghitung jumlah setiap sentimen
positif = sentiment_counts.get('Positif', 0) # Mengambil jumlah sentimen positif
negatif = sentiment_counts.get('Negatif', 0) # Mengambil jumlah sentimen negatif

labels = ['Positif ((positif))', 'Negatif ((negatif))'] # Membuat label untuk pie chart
sizes = [positif, negatif] # Menentukan ukuran setiap bagian pie
colors = ['#2196F3', '#8BC34A'] # Menentukan warna untuk setiap bagian pie

plt.figure(figsize=(8, 6)) # Membuat figure dengan ukuran 8x6
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=120) # Membuat pie chart
plt.axis('equal') # Menastikan pie berbentuk lingkaran
plt.title('Distribusi Sentimen') # Menambahkan judul grafik
plt.show() # Menampilkan grafik

MENAMPILKAN DATA FRAME
[] # @title MENAMPILKAN DATA FRAME
columns = ['full_text', 'Sentimen'] # Mendefinisikan kolom yang akan digunakan
df = pd.DataFrame(data, columns=columns).iloc[0:2115] # Membuat DataFrame dengan 540 baris pertama dari data
df.head(5) # Menampilkan 5 baris pertama dari DataFrame

CASE FOLDING
[] # @title CASE FOLDING
def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower() # Meneriksa apakah input adalah string
        return lowercase_text # Mengubah teks menjadi huruf kecil
    else:
        return text # Mengembalikan input asli jika bukan string
df['case_folding'] = df['full_text'].apply(case_folding) # Menerapkan fungsi case_folding pada kolom 'full_text'
df.head(5) # Menampilkan 5 baris pertama dari DataFrame

CLEANSING
[] # @title CLEANSING
def clean_text(tweet):
    tweet = re.sub(r'https?://\S+|\S+', '', tweet) # Menghapus URL
    tweet = re.sub(r'<.*>', '', tweet) # Menghapus tag HTML
    emoji_pattern = re.compile("["
        u'\U0001F600-\U0001F64F' # emoticons
        u'\U0001F300-\U0001F3FF' # simbol & pictogram
        u'\U0001F500-\U0001F5FF' # simbol transportasi & peta
        u'\U0001F6C0-\U0001F6FF' # bendera (iOS)
        u'\U00002700-\U000027BF'
        u'\U000024C0-\U000024FF"
    ]) # flag unicode
    tweet = emoji_pattern.sub('', tweet) # Menghapus emoji
    tweet = re.sub(r'@w+', '', tweet) # Menghapus username
    tweet = re.sub(r'#+', '', tweet) # Menghapus angka
    tweet = re.sub(r'[\s_]+', ' ', tweet) # Normalisasi karakter berulang
    tweet = re.sub(r'[\W_]+', '', tweet) # Menghapus simbol kecuali underscore
    tweet = re.sub(r'^\s+', '', tweet).strip() # Menghapus spasi berlebih
    return tweet # Mengembalikan teks yang sudah dibersihkan
df['cleaning'] = df['case_folding'].apply(clean_text) # Menerapkan fungsi clean_text ke kolom 'case_folding'
df.head(5) # Menampilkan 5 baris pertama dari DataFrame

STOPWORD REMOVAL
[] # @title STOPWORD REMOVAL
stop_words = stopwords.words('Indonesian') # Mengambil daftar stopwords bahasa Indonesia dari NLTK
additional_stopwords = [
    'ya', 'aye', 'aja', 'aga', 'aya', 'yah', 'ada', 'aku', 'kamu', 'dia', 'manusia',
    'kita', 'saya', 'halo', 'itu', 'sini', 'tuh', 'tuh', 'deh', 'dong', 'kay', 'lgsa',
    'ken', 'pun', 'ah', 'kok', 'mni', 'mah', 'blm', 'tdk', 'di', 'di', 'tp', 'ngelihat',
    'lwk', 'trs', 'egn', 'sy', 'lg', 'dg', 'km', 'sni', 'pd', 'ak', 'sm', 'agt', 'nder',
    'kl', 'krn', 'aj', 'eng', 'amp', 'dm', 'ga', 'lu', 'low', 'lo', 'ni', 'abis', 'na',
    'sini', 'big', 'nah', 'kha', 'kiau', 'ng', 'td', 'udah', 'em', 'dot', 'perasaan',
    'nti', 'gini', 'kni', 'yuk', 'sennya', 'krng', 'skg', 'sya', 'gmn', 'tys', 'udah',
    'ong', 'sewa', 'ntn', 'gitu', 'spt', 'mmn', 'oto', 'tntst', 'tuh', 'kaki', 'pals',
    'kan', 'big', 'skrg', 'lwk', 'udah', 'kni', 'dini', 'mgg', 'mlm', 'padi', 'pagsi', 'gema',
    'trem', 'kita', 'yuk', 'ah', 'yuk', 'nek', 'jgn', 'dg', 'di', 'ng', 'ni', 'di', 'tpi', 'low']
stop_words.extend(additional_stopwords) # Menambahkan stopwords tambahan ke dalam daftar
def remove_stopwords(text):
    words = text.split() # Membagi teks menjadi kata-kata
    cleaned_text = ' '.join(word for word in words if word not in stop_words) # Menghapus stopwords dan menggabungkan kata-kata kembali
    return cleaned_text
df['stopword_removal'] = df['cleaning'].apply(remove_stopwords) # Menerapkan fungsi remove_stopwords pada kolom 'cleaning'
df.head(5) # Menampilkan 5 baris pertama dari DataFrame

STEMMING
[] # @title STEMMING
factory = StemmerFactory() # Membuat objek StemmerFactory
stemmer = factory.create_stemmer() # Membuat objek stemmer dari factory
def stem_text(text):
    stemmed_text = ' '.join([stemmer.stem(word) for word in text.split()]) # Melakukan stemming pada setiap kata dan menggabungkannya kembali
    return stemmed_text
df['stemming'] = df['stopword_removal'].apply(stem_text) # Menerapkan fungsi stem_text pada kolom 'stopword_removal'
df.head(5) # Menampilkan 5 baris pertama dari DataFrame

DOWNLOAD CSV HASIL PRE-PROCESSING
[] # @title DOWNLOAD CSV HASIL PRE-PROCESSING
df.to_csv('HASIL-PRE.csv', encoding='utf8', index=False)
```

## Lampiran 4 Code *Word Cloud*

```

VISUALISASI WORD CLOUD
[ ] # @title VISUALISASI WORD CLOUD
import matplotlib.pyplot as plt # Import plotting library
from wordcloud import WordCloud # Import WordCloud for visualization
from nltk.corpus import stopwords, wordnet # Import NLTK corpus tools
import pandas as pd # Import pandas for data manipulation
import nltk # Import NLTK library
nltk.download('stopwords') # Download stopwords dataset
nltk.download('wordnet') # Download WordNet dataset

kunci_utama = ['quick', 'count', 'quick count'] # Define main keywords to remove

def get_synonyms(word): # Function to get synonyms
    synonyms = set()
    for syn in wordnet.synsets(word):
        for lemma in syn.lemmas():
            synonyms.add(lemma.name().replace('_', ' '))
    return synonyms

all_stopwords = set(stopwords.words('indonesian')) # Get Indonesian stopwords
for key in kunci_utama: # Add synonyms of main keywords
    all_stopwords.update(get_synonyms(key))
all_stopwords.update(kunci_utama) # Add main keywords to stopwords

def clean_text(text): # Function to clean text
    words = text.split()
    return ' '.join([word for word in words if word.lower() not in all_stopwords])

positif_text = ' '.join(df[df['Sentimen'] == 'Positif']['stemming']) # Concatenate positive texts
negatif_text = ' '.join(df[df['Sentimen'] == 'Negatif']['stemming']) # Concatenate negative texts

positif_text_clean = clean_text(positif_text) # Clean positive text
negatif_text_clean = clean_text(negatif_text) # Clean negative text

def generate_wordcloud(text): # Function to generate word cloud
    return WordCloud(width=800, height=400, background_color='white', stopwords=all_stopwords).generate(text)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(20, 10)) # Create subplot

wordcloud_positif = generate_wordcloud(positif_text_clean) # Generate positive word cloud
ax1.imshow(wordcloud_positif, interpolation='bilinear') # Display positive word cloud
ax1.axis('off') # Turn off axis
ax1.set_title('Word Cloud Sentimen Positif') # Set title for positif

wordcloud_negatif = generate_wordcloud(negatif_text_clean) # Generate negative word cloud
ax2.imshow(wordcloud_negatif, interpolation='bilinear') # Display negative word cloud
ax2.axis('off') # Turn off axis
ax2.set_title('Word Cloud Sentimen Negatif') # Set title for negatif

plt.tight_layout() # Adjust layout
plt.show() # Display the plot

```

## Lampiran 5 Code Pembobotan TF-IDF

```

PEMBOBOTAN TF-IDF
[ ] # @title PEMBOBOTAN TF-IDF
import pandas as pd # Mengimpor library pandas
import numpy as np # Mengimpor library numpy
from sklearn.feature_extraction.text import TfidfVectorizer # Mengimpor TfidfVectorizer dari sklearn

df = pd.read_csv('HASIL-PRE.csv') # Membaca file CSV
documents = df['stemming'].tolist() # Mengambil data stemming sebagai list
sentiments = df['Sentimen'].tolist() # Mengambil data sentimen sebagai list
N = len(documents) # Menghitung jumlah dokumen

vectorizer = TfidfVectorizer() # Membuat objek TfidfVectorizer
tfidf_matrix = vectorizer.fit_transform(documents) # Melakukan fit dan transform pada dokumen
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=vectorizer.get_feature_names_out()) # Membuat DataFrame dari matriks TF-IDF

total_vocabulary = len(vectorizer.get_feature_names_out()) # Menghitung total kosakata unik
print(f" Total Vocabulary: {total_vocabulary}") # Mencetak total kosakata

print("\n Term Frequency-Inverse Document Frequency (TF-IDF):") # Mencetak judul hasil
for i, (row, col) in enumerate(zip(tfidf_matrix.nonzero()[0][:40], tfidf_matrix.nonzero()[1][:40])): # Loop untuk 40 elemen pertama
    print(f" ({row}, {col})\t{tfidf_matrix[row, col]}") # Mencetak nilai TF-IDF
print("\n ...") # Mencetak pemisah
for i, (row, col) in enumerate(zip(tfidf_matrix.nonzero()[0][-3:], tfidf_matrix.nonzero()[1][-3:])): # Loop untuk 3 elemen terakhir
    print(f" ({row}, {col})\t{tfidf_matrix[row, col]}") # Mencetak nilai TF-IDF

DOWLOAD EXCEL HASIL TF-IDF
[ ] # @title DOWLOAD EXCEL HASIL TF-IDF
import pandas as pd # Mengimpor library pandas
import numpy as np # Mengimpor library numpy
from sklearn.feature_extraction.text import TfidfVectorizer # Mengimpor TfidfVectorizer dari sklearn

df = pd.read_csv('HASIL-PRE.csv') # Membaca file CSV hasil preprocessing
documents = df['stemming'].tolist() # Mengambil data stemming sebagai list
sentiments = df['Sentimen'].tolist() # Mengambil data sentimen sebagai list
N = len(documents) # Menghitung jumlah dokumen

vectorizer = TfidfVectorizer() # Membuat objek TfidfVectorizer
tfidf_matrix = vectorizer.fit_transform(documents) # Melakukan fit dan transform pada dokumen
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=vectorizer.get_feature_names_out()) # Mengubah hasil menjadi DataFrame
tfidf_df.to_excel('HASIL-TFIDF.xlsx', index=False) # Menyimpan DataFrame ke file Excel

```

## Lampiran 6 Code Split Data

```

v SPLIT DATA

[ ] # @title SPLIT DATA
from sklearn.feature_extraction.text import TfidfVectorizer # Mengimpor TfidfVectorizer dari sklearn
from sklearn.model_selection import train_test_split # Mengimpor train_test_split dari sklearn
import matplotlib.pyplot as plt # Mengimpor library matplotlib

tfidf_vectorizer = TfidfVectorizer() # Membuat objek TfidfVectorizer
X = tfidf_vectorizer.fit_transform(df['stemming']) # Melakukan fit dan transform pada kolom 'stemming'
y = df['Sentimen'] # Mengambil kolom 'Sentimen' sebagai target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) # Membagi data menjadi set training dan testing
total_samples = X.shape[0] # Menghitung total sampel
train_samples = X_train.shape[0] # Menghitung jumlah sampel training
test_samples = X_test.shape[0] # Menghitung jumlah sampel testing
fig, ax = plt.subplots(figsize=(10, 6)) # Membuat figure dan axes

bars = ax.bar(['Total Sampel', 'Sampel Training', 'Sampel Testing'], # Membuat bar chart
              [total_samples, train_samples, test_samples], # Data untuk setiap bar
              color=['#4682B4', '#80C4DE', '#80E0E6']) # Warna untuk setiap bar

for bar in bars:
    height = bar.get_height() # Loop untuk setiap bar
    ax.text(bar.get_x() + bar.get_width()/2., height, # Mendapatkan tinggi bar
            f'{height}', # Menambahkan teks di atas bar
            ha='center', va='bottom') # Teks yang ditambahkan (jumlah sampel)
                                # Pengaturan posisi teks

ax.set_ylabel('Jumlah Sampel') # Menambahkan label sumbu y
ax.set_title('Distribusi Sampel dalam Dataset') # Menambahkan judul grafik

ax.text(1, train_samples/2, f'{train_samples/total_samples:.1%}', # Menambahkan persentase pada bar training
        ha='center', va='center') # Pengaturan posisi teks
ax.text(2, test_samples/2, f'{test_samples/total_samples:.1%}', # Menambahkan persentase pada bar testing
        ha='center', va='center') # Pengaturan posisi teks
plt.tight_layout() # Mengatur layout agar tidak tumpang tindih
plt.show() # Menampilkan grafik

```

## Lampiran 7 Code Klasifikasi

```

v EVALUASI MODEL NAIVE BAYES DAN VISUALISASI CONFUSION MATRIX

[ ] # @title EVALUASI MODEL NAIVE BAYES DAN VISUALISASI CONFUSION MATRIX
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score # Mengimpor metrik evaluasi
from sklearn.naive_bayes import MultinomialNB # Mengimpor MultinomialNB classifier
import seaborn as sns # Mengimpor library Seaborn untuk visualisasi data
import matplotlib.pyplot as plt # Mengimpor library Matplotlib untuk plotting

nb_classifier = MultinomialNB() # Inisialisasi classifier Naive Bayes
nb_classifier.fit(X_train, y_train) # Melatih classifier dengan data training
y_pred = nb_classifier.predict(X_test) # Melakukan prediksi pada data testing
conf_matrix = confusion_matrix(y_test, y_pred) # Menghitung confusion matrix

sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', # Membuat heatmap dari confusion matrix
            xticklabels=['NEGATIF', 'POSITIF'], # Memberikan label pada sumbu x
            yticklabels=['NEGATIF', 'POSITIF']) # Memberikan label pada sumbu y

plt.xlabel('Predicted') # Memberikan label pada sumbu x
plt.ylabel('Actual') # Memberikan label pada sumbu y
plt.title('Confusion Matrix') # Memberikan judul pada plot
plt.show() # Menampilkan plot

accuracy = accuracy_score(y_test, y_pred) # Menghitung akurasi model
print("\nAccuracy:", accuracy) # Menampilkan akurasi
print("\nClassification Report:") # Menampilkan judul laporan klasifikasi
print(classification_report(y_test, y_pred, zero_division=1)) # Menampilkan laporan klasifikasi lengkap

```

## Lampiran 8 Lembar Tindak Lanjut

### Dokumen Revisi dan Tindak Lanjut Peserta Ujian Sidang Hasil


Nama: Aditya Pranata

NIM: 2011102441052


No	Nama Dosen	Permintaan Revisi	Tindak lanjut	Halaman
1	Naufal Azmi Verdikha, S.Kom., M.Eng	Judul perlu direvisi, karena metode penelitian yang dimaksud bukan untuk mendeteksi atau mencari hasil quick count	Telah di rubah dengan judul baru sesuai dengan penelitian	i-iii
2	Naufal Azmi Verdikha, S.Kom., M.Eng	Penulisan belum sesuai dengan pedoman	Telah di rubah sesuai dengan pedoman	1-3
3	Naufal Azmi Verdikha, S.Kom., M.Eng	Belum ada deskripsi urgensi penelitian. Urgensi ditulis untuk melihat fokus penelitian dan pentingnya penelitian tersebut dilakukan.	Sudah ditambahkan urgensi dari penelitian yang diminta	1-4
4	Naufal Azmi Verdikha, S.Kom., M.Eng	Rumusan masalah masih di tingkat D3, belum sampai di tahap analisis. Rumusan masalah perlu melibatkan unsur metode, objek, dan tolak ukur penelitian	Telah dirubah dan diperbaiki pada rumusan masalah	2
5	Naufal Azmi Verdikha, S.Kom., M.Eng	Setiap Tahapan penelitian perlu dijelaskan perlangkah-langkah dari awal hingga menghasilkan tahapan yang dituju.	Sudah ditambahkan setiap tahapan penelitian pada bab 2	4-10
6	Naufal Azmi Verdikha, S.Kom., M.Eng	Penentuan rasio data split perlu ada landasan kuat	Sudah ditambahkan referensi landasan kuat pada penentuan rasio split data	8
8	Rudiman, S.Kom., M.Sc	Naskah harus dibuat minimal sampai BAB 3	Telah dibuat naskah sampai dengan BAB 3	12-18

Menyetujui


Penguji 1

  
Naufal Azmi Verdikha, S.Kom., M.Eng  
NIDN 1114048801

Penguji 2

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

## Lampiran 9 Surat Izin Penelitian

 <b>UMKT</b> Program Studi <b>Teknik Informatika</b> Fakultas Sains dan Teknologi	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 : 055-002/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:

No	Nama	NIM
1	Ricky Albin Pranata	2011102441017
2	Aditya Pranata	2011102441052


Melakukan penelitian dengan pengambilan data sekunder di Media Sosial Twitter data yang diambil yaitu sentimen mengenai quick count.


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

*Wassalamu'alaikum Warrahmatullahi Wabarrakatuh*

Samarinda, 19 Dzulhijjah 1445 H  
26 Juni 2024 M

Program Studi S1 Teknik Informatika

  
Hausyah, S.Kom., M.TI  
N. 1118019203



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

## Lampiran 10 Kartu Kendali Bimbingan

### KARTU KENDALI BIMBINGAN LAPORAN KARYA ILMIAH

Nama : Aditya Pranata  
 NIM : 2011102441052  
 Nama Dosen Pembimbing : Rudiman, S.Kom., M.Sc  
 Judul Penelitian : Metode Pembobotan TF-IDF untuk Klasifikasi Teks Quick Count Pemilihan Presiden Indonesia 2024 pada X Twitter Menggunakan Metode Naive Bayes

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1	07/02/2024	Persetujuan bimbingan dengan topik penelitian	h
2	14/02/2024	Mencari permasalahan topik yang diteliti dan objek	h
3	22/02/2024	Malutuban tahap perencanaan data di python googlecolab	h
4	29/02/2024	Menentukan judul dan mulai perencanaan latar belakang yang ditetapkan oleh dosen pembimbing	h
5	09/03/2024	Pengayaan latar belakang harus mengikuti urutan yang telah ditetapkan oleh dosen pembimbing	h
6	13/03/2024	Malutuban revisi yang diarahkan dosen pembimbing	h
7	18/03/2024	Dosen pembimbing membantu beberapa saran terhadap konsep pengajuan judul ke prodi	h
8	27/03/2024	membantu beberapa revisi terhadap latar belakang, rumusan masalah, tujuan, dan manfaat	h
9	08/04/2024	Masuk ke bab 2 menggunakan strategi yang telah dibarek dosen pembimbing	h
10	24/04/2024	membantu masukan dan revisi pada Bab 2	h
11	26/06/2024	Dosen pembimbing memberi masukan Bab 3	h
12	09/07/2024	beberapa saran dan masukan dari dosen pembimbing terkait Seminar hasil	h
13	10/07/2024	membantu revisi terkait jurnal yang akan di submit	h
14	15/07/2024	Malutuban revisi Bab 3 yang diarahkan dosen pembimbing	h

Dosen Pembimbing

(Rudiman, S.Kom., M.Sc.)



(S.Kom., M.TI)



# SKRIPSI ADITYA PRANATA

by S1 Teknik Informatika Universitas Muhammadiyah Kalimantan Timur



---

**Submission date:** 16-Jul-2024 02:26PM (UTC+0800)

**Submission ID:** 2417653969

**File name:** SKRIPSI-ADITYA\_PRANATA\_-\_ADITYA\_PRANATA.docx (948.62K)

**Word count:** 6186

**Character count:** 39062

## SKRIPSI ADITYA PRANATA

### ORIGINALITY REPORT

<b>13%</b>	<b>10%</b>	<b>7%</b>	<b>5%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

### PRIMARY SOURCES

<b>1</b>	Untung Surapati, Ali Yafi Zulkarnain. "Implementasi Metode Naïve Bayes Untuk Mendeteksi Hate Speech Pada Twitter", INTECOMS: Journal of Information Technology and Computer Science, 2023 Publication	<b>1%</b>
<b>2</b>	Submitted to Sriwijaya University Student Paper	<b>1%</b>
<b>3</b>	123dok.com Internet Source	<b>1%</b>
<b>4</b>	ejournal.ust.ac.id Internet Source	<b>1%</b>
<b>5</b>	dspace.umkt.ac.id Internet Source	<b>1%</b>
<b>6</b>	media.neliti.com Internet Source	<b>&lt;1%</b>
<b>7</b>	journal.maranatha.edu Internet Source	<b>&lt;1%</b>
<b>8</b>	Submitted to Academic Library Consortium Student Paper	