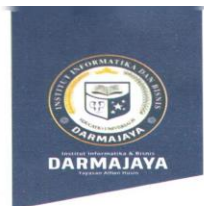


LAMPIRAN

Lampiran 1 Surat Keputusan



SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK. 0543/DMJ/DFIK/BAAK/X-23
 Tentang
Dosen Pembimbing Skripsi
Semester Ganjil TA.2023/2024
Program Studi S1 Teknik Informatika

REKTOR IIB DARMAJAYA

- Memperhatikan :** 1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam **Skripsi**.
- Menimbang :** 1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan **Dosen Pembimbing Skripsi**.
- Mengingat :** 2. Bahwa untuk maksud tersebut dipandang perlu menerbitkan Surat Keputusan Rektor.
- Mengingat :** 1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
 2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
 6. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
 7. STATUTA IBI Darmajaya
 8. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi
 6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.
- Menetapkan**
- Pertama :** Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Teknik Informatika.
- Kedua :** Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.
- Ketiga :** Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma pengajian dan honorarium IBI Darmajaya.
- Keempat :** Surat Keputusan ini berlaku sejak tanggal ditetapkan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung
 Pada tanggal : 16 Oktober 2023
 a.n. Rektor IIB Darmajaya,
 Dekan Fakultas Ilmu Komputer


 Dr. Suredi, S.Kom., M.T.I
 NIK. 00590203

1. Kepala Program Studi S1 Teknik Informatika
2. Yang bersangkutan
3. Arsip

Lampiran : Surat Keputusan Rektor IIB Darmajaya
 Nomor : SK 0543/DMI/DFIK/BAK/X-23
 Tanggal : 16 Oktober 2023
 Perihal : Pembimbing Penulisan Skripsi Semester Ganjil TA. 2023/2024
 Program Studi Strata Satu (S1) Teknik Informatika

Judul Skripsi Dan Dosen Pembimbing Skripsi Semester Ganjil TA. 2023/2024
Program Studi Strata Satu (S1) Teknik Informatika

No	NAMA	NPM	JUDUL	PEMBIMBING
1	Yoga Budiman	2011010070	Pengenaln Informasi Pakaian dan Alat Tradisional Kebudayaan Pesisir Barat Menggunakan Virtual Reality (VR) Berbasis Android (Studi Kasus : Lamban Balak)	R.Z. Abdul Aziz, M.T., Ph.D
2	Muhammad Alvian	2011010071	Implementasi Pembelajaran Bacaan Tajwid dan Iqro Teknologi Augmented Reality	
3	Latifah Nabillah	2011010045	Diagnosis Penyakit Daun Mawar Menggunakan Pendekatan Deep Learning	Nisar, M.T.
4	Indah Amella	2011010023	Diagnosa Penyakit Daun Tomat Menggunakan Metode VGG-19	
5	Putri Diah Agustina	2011010065	Implementasi Sistem Rekomendasi Pemilihan Mata Kuliah Peminatan Mahasiswa Baru Menggunakan Algoritma Support Vector Machine Berbasis Website (Studi Kasus : Teknik Informatika IIB Darmajaya)	
6	Deani Intan Sari	2011010011	Analisis Sentimen Terhadap Penggunaan Aplikasi EdLink Menggunakan Algoritma Support Vector Machine	Suhendro Yusuf Irianto, M.Kom., Ph.D
7	Dila Mutlara	2011010005	Deteksi Cyberbullying Pada Komentar Youtube Dengan Metode Naïve Bayes	
8	Welda Syahfira	2011010096	Penerapan Metode Knowledge Based Recommendation dalam Menentukan Perkiraan Mahasiswa Magang Berdasarkan Kesesuaian Jurusan dan Kebutuhan Bisnis UMKM	
9	Irfan Adigumanto	1911010079	Penerapan Sistem File Encryption pada Sistem Operasi Linux untuk Menjaga Kerahasiaan Data Pengguna	DR. Chairani, S.Kom., M.Eng
10	Adella Marsha Safira	2011010016	Sistem Pemilihan Sampah Dengan Scale Invariant Feature Transform (SIFT)	
11	Bhagus Adi Pratama	2011010134	Pengembangan Aplikasi Deteksi Ekspresi Wajah Secara Real Time Menggunakan Convolution Neural Network	
12	Irfham Febriansyah	2011010121	Implementasi Algoritma K-Means Untuk Klastering Destinasi Wisata Di Kabupaten Pesawaran Berbasis Web (Studi Kasus: Dinas Pariwisata Kabupaten Pesawaran)	
13	Muhammad Fattan Abdul Aziz	1911010110	Aplikasi Pengenalan Kebudayaan Lampung Menggunakan Metode Augmented Reality Berbasis Android	DR. Sri Lestari, S.Kom., M.Cs
14	Hawali Nur Aqil	2011010026	Sistem Informasi dan Monitoring Industry Kreatif (SIMOKRAF) pada Kabupaten Pesawaran	
15	Muhammad Fattan Abdul Aziz	1911010110	Aplikasi Pengenalan Kebudayaan Lampung Menggunakan Metode Augmented Reality Berbasis Android	
16	Chandry Vivia Khoirunnisa	2011010105	Sistem Rekomendasi Program Studi Menggunakan Machine Learning Untuk Siswa Sekolah Menengah Atas (Studi Kasus: Institut Informatika dan Bisnis Darmajaya)	
17	Riko Dwi Saputra	2011010079	Pengembangan Permainan Edukasi Interaktif Untuk Anak Pada Mata Pelajaran Matematika Berbasis Mobile Menggunakan Metode ADDE	DR. Yuni Arkhiansyah, M.Kom.
18	Audina Tazkia	2011010090	Implementasi Metode Naive Bayes Classifier Dalam Menganalisis Sentimen Pelanggan Terhadap Mlie Gacooan Pada Instagram	

Lampiran 2 Proses Pelabelan Manual



Proses Pelabelan Manual

1. Crawling Komentar Video Youtube

a. Import Library

```
import pandas as pd
from googleapiclient.discovery import build
```

b. Fungsi untuk crawling komentar

```
def video_comments(video_id):
    replies = []
    youtube = build('youtube', 'v3',
                    developerKey=api_key)

    video_response =
    youtube.commentThreads().list(part='snippet,replies',
                                   videoId=video_id).execute()

    while video_response:
        for item in video_response['items']:
```

```

        published =
item['snippet']['topLevelComment']['snippet']['publ
ishedAt']
        user =
item['snippet']['topLevelComment']['snippet']['auth
orDisplayName']
        comment =
item['snippet']['topLevelComment']['snippet']['text
Display']
        likeCount =
item['snippet']['topLevelComment']['snippet']['like
Count']

        replies.append([published, user, comment,
likeCount])

        replycount =
item['snippet']['totalReplyCount']
        if replycount>0:
            for reply in item['replies']['comments']:
                published =
reply['snippet']['publishedAt']
                user =
reply['snippet']['authorDisplayName']
                repl = reply['snippet']['textDisplay']
                likeCount = reply['snippet']['likeCount']
                replies.append([published, user, repl,
likeCount])
            if 'nextPageToken' in video_response:
                video_response =
youtube.commentThreads().list(
                    part = 'snippet,replies',
                    pageToken =
video_response['nextPageToken'],
                    videoId = video_id
                ).execute()
            else:
                break
        return replies

```

c. Jalankan Proses Crawling

```
api_key = '****SyA9G4vsDDRtZsYz23sveQzBs*****'
```

```
video_id = "C8NvLMfJagE" #isikan dengan kode / ID
video
comments = video_comments(video_id)
comments
```

d. Simpan Hasil Crawling ke file CSV

```
df.to_csv('1.data youtube crawling.csv',
index=False)
```

2. Preprocessing

a. Import file hasil crawling

```
import pandas as pd
import numpy as np
#import file yang akan di gunakan
from google.colab import files
uploades = files.upload()
#untuk menampilkan isi data
data = pd.read_csv('1.data youtube crawling.csv')
data.head(10)
#menghilangkan data yang duplikat
data.drop_duplicates(subset = "textDisplay", keep =
'first', inplace = True)
data.info()
#dataframe mengelompokkan
df = pd.DataFrame(data['textDisplay'])
df.head()
```

b. Cleaning

```
import re
import string
import nltk

def remove_URL(youtube):
    url = re.compile(r'https?://\S+|www\.\S+')
    return url.sub(r'', youtube)

def remove_html(youtube):
    html = re.compile(r'<.*?>')
    return html.sub(r'', youtube)

def remove_emoji(youtube):
    emoji_pattern = re.compile("[
```

```

u"\U0001F600-\U0001F64F"
u"\U0001F300-\U0001F5FF"
u"\U0001F680-\U0001F6FF"
u"\U0001F1E0-\U0001F1FF"
        "]" + "", flags=re.UNICODE)
return emoji_pattern.sub(r'', youtube)

def remove_numbers(youtube):
    youtube = re.sub(r'\d+', '', youtube)
    return youtube

def remove_symbols(youtube):
    youtube = re.sub(r'^[a-zA-Z0-9\s]', '',
youtube) #menghapus semua simbol
    return youtube

df['cleasing'] = df['textDisplay'].apply(lambda x:
remove_URL(x))
df['cleasing'] = df['cleasing'].apply(lambda x:
remove_html(x))
df['cleasing'] = df['cleasing'].apply(lambda x:
remove_emoji(x))
df['cleasing'] = df['cleasing'].apply(lambda x:
remove_symbols(x))
df['cleasing'] = df['cleasing'].apply(lambda x:
remove_numbers(x))
df.head(10)

```

c. Case Folding

```

def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower()
        return lowercase_text
    else:
        return text
df['case_folding'] =
df['cleasing'].apply(case_folding)
df.head()

```

d. Tokenization

```

def tokenize(text):
    tokens = text.split()

```

```
    return tokens
```

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

e. Filtering/Stopword Removal

```
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('indonesian')
def remove_stopwords(text):
    return [word for word in text if word not in
            stop_words]

df['filtering/stopword_removal'] =
df['tokenize'].apply(lambda x: remove_stopwords(x))
df.head()
```

f. Steaming Data

```
!pip install Sastrawi
from Sastrawi.Stemmer.StemmerFactory import
StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

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

df['steaming_data'] =
df['filtering/stopword_removal'].apply(lambda x: '
'.join(stem_text(x)))
df.head()
```

g. Simpan Hasil Preprocessing ke file CSV

```
df.to_csv('2.Data_Preprocessing_Hasil.csv')
```

3. Pelabelan Vader Lexicon

a. Menampilkan Dataset

```
#import library
import pandas as pd
```



```
import numpy as np
#import file yang akan di gunakan
from google.colab import files
uploades = files.upload()
#untuk menampilkan isi data
df = pd.read_csv('2.Data_Preprocessing_Hasil.csv')
df.head(10)
```

b. Proses

```
!pip install nltk
import nltk
from nltk.sentiment import
SentimentIntensityAnalyzer
import pandas as pd
nltk.download('punkt')
nltk.download('vader_lexicon')

#pelebelan
data = SentimentIntensityAnalyzer()
df['steaming_data'] =
df['steaming_data'].fillna('')
labels = []
scores = []
for text in df['steaming_data']:
    sentiment_scores = data.polarity_scores(text)
    compound_score = sentiment_scores['compound']
    scores.append(compound_score)
    if compound_score > 0:
        label = 'non_cyberbullying'
    else:
        label = 'cyberbullying'
    labels.append(label)
df['sentimen_score'] = scores
df['sentimen'] = labels
data = ['steaming_data', 'sentimen_score',
'sentimen']
data = df[data]
data.head(10)
#simpan file
df.to_csv("3.Hasil_Pelabelan_Phyton.csv")
```

c. Visualisasi Sentimen

```

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

sentimen_count = data['sentimen'].value_counts()
sns.set_style('whitegrid')
colors = {'non_cyberbullying': 'lightskyblue',
          'cyberbullying': 'red'}

fig, ax = plt.subplots(figsize=(6,4))
ax = sns.barplot(x=sentimen_count.index,
                y=sentimen_count.values, palette=colors.values())
plt.title('Jumlah Data Pelebelan Vader Lexicon',
          fontsize=14, pad=20)
plt.xlabel('Class Sentimen', fontsize=12)
plt.ylabel('Jumlah Komentar', fontsize=12)

for i, count in enumerate(sentimen_count.values):
    ax.text(i, count+0.10, str(count), ha='center',
            va='bottom')
plt.show()

```

d. WordCloud

```

#WORDCLOUD CYBERBULLTING
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Contoh data teks
all_text_s0 = ''.join(word for word in
data_cyberbullying["steaming_data"])
# Kata-kata yang ingin dihilangkan
unwanted_words = ['yg', 'lu', 'aja', 'ga', 'gak',
                  'gk', 'nya', 'ya', 'gw', 'sih', 'klo', 'pake',
                  'nih', 'kalo', 'udah', 'org', 'si', 'tu', 'kali',
                  'kek', 'bgt', 'dpt', 'bar', 'tau', 'kak']
# Menghilangkan kata-kata yang tidak diinginkan
for word in unwanted_words:

```

```

    all_text_s0 = all_text_s0.replace(word, '')
# Membuat WordCloud
wordcloud = WordCloud(colormap='Reds', width=1000,
height=1000, mode='RGBA',
background_color='white').generate(all_text_s0)
# Menampilkan WordCloud
plt.figure(figsize=(9, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("WordCloud Cyberbullying")
plt.show()
# Mengambil kata-kata yang paling sering muncul
setelah pembersihan
word_frequencies =
wordcloud.process_text(all_text_s0)
top_words = sorted(word_frequencies.items(),
key=lambda x: x[1], reverse=True)[:30]
print("Kata Cyberbullying yang paling sering
muncul:")
for word, freq in top_words:
    print(f"{word}: {freq}")
# Mengambil kata-kata yang paling sering muncul
setelah pembersihan
word_frequencies =
wordcloud.process_text(all_text_s0)
top_words = sorted(word_frequencies.items(),
key=lambda x: x[1], reverse=True)[:10]
# Membuat grafik
words, frequencies = zip(*top_words)
plt.figure(figsize=(9, 6))
plt.barh(words, frequencies, color='salmon')
plt.xlabel('Frekuensi')
plt.title('Kata-Kata yang Sering Muncul dalam
Cyberbullying')
plt.show()

```

```

#WORDCLOUD NON-CYBERBULLYING
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import pandas as pd

```

```

# Ubah nilai NaN menjadi string kosong
df['steaming_data'] =
df['steaming_data'].fillna('')
df_n = df[df['sentimen'] == 'non_cyberbullying'] #
Filter DataFrame hanya untuk label Non-
Cyberbullying
# Gabungkan semua teks non_cyberbullying dalam satu
string
all_text_s0 = ' '.join(str(word) for word in
df_n["steaming_data"])
# Kata-kata yang ingin dihilangkan
unwanted_words = ['yg', 'lu', 'aja', 'ga', 'gak',
'gk', 'nya', 'ya', 'gw', 'sih', 'klo', 'pake',
'nih', 'kalo', 'udah', 'org', 'si', 'tu', 'kali',
'kek', 'bgt', 'dpt', 'bar', 'tau', 'kak']
# Membersihkan teks dari kata-kata yang tidak
diinginkan
cleaned_text = ' '.join(word for word in
all_text_s0.split() if word.lower() not in
unwanted_words)
# Buat WordCloud dari teks non_cyberbullying
wordcloud = WordCloud(colormap='Blues', width=1000,
height=1000, mode='RGBA',
background_color='white').generate(cleaned_text)
# Tampilkan WordCloud
plt.figure(figsize=(9, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("WordCloud Non-Cyberbullying")
plt.show()
# Mengambil kata-kata yang paling sering muncul
setelah pembersihan
word_frequencies =
wordcloud.process_text(cleaned_text)
top_words = sorted(word_frequencies.items(),
key=lambda x: x[1], reverse=True)[:30]
print("Kata Non-Cyberbullying yang paling sering
muncul:")
for word, freq in top_words:
    print(f"{word}: {freq}")

```

4. Pembagian Data Training dan Testing

```
#membagi data menjadi data training dan testing
dengan test_size = 0.20 dan random state nya 0
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(data['steaming_data'],
data['klasifikasi'],
                                test_size = 0.20,
                                random_state = 0)

print("Data training (X_train):")
print(X_train.head(10))
print("\nData testing (X_test):")
print(X_test.head(10))
```

5. Pembobotan TF-IDF

```
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.model_selection import train_test_split

X_train = X_train.replace(np.nan, '', regex=True)

# Inisialisasi TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()

# Melakukan transformasi TF-IDF pada data latih dan
data uji
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_test = tfidf_vectorizer.transform(X_test)

print(tfidf_train)
print(tfidf_test)
```

6. Analisis Hasil

```
from sklearn.naive_bayes import MultinomialNB

nb = MultinomialNB()
nb.fit(tfidf_train, y_train)

from sklearn.model_selection import cross_val_predict
```

```

from sklearn.metrics import accuracy_score,
classification_report, confusion_matrix
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import
TfidfVectorizer
from sklearn.model_selection import cross_val_predict
from sklearn.model_selection import cross_val_score

# Inisialisasi dan latih model Naive Bayes
clf = MultinomialNB()
clf.fit(tfidf_train, y_train)

# Prediksi pada data uji
predicted = clf.predict(tfidf_test)
# Menghitung akurasi
accuracy = accuracy_score(y_test, predicted)
# Menampilkan confusion matrix dan classification
report
print("Akurasi Model Naive Bayes : ", accuracy)
print(f'confusion_matrix:\n {confusion_matrix(y_test,
predicted)}')
print('=====\n')
print('\nClassification Report:\n',
classification_report(y_test, predicted,
zero_division=0))

cross_val_scores = cross_val_score(clf, tfidf_train,
y_train, cv=5, scoring='accuracy')
average_accuracy_cross_val = cross_val_scores.mean()
print("Akurasi Cross-Validation:
{:.2%}".format(average_accuracy_cross_val))

```

7. Metrik Untuk 3 Skenario

```

import matplotlib.pyplot as plt

# Data
scenarios = ['Skenario ke-1', 'Skenario ke-2',
'Skenario ke-3']
precision_cyberbullying = [64, 86, 80]

```

```
precision_non_cyberbullying = [79, 100, 100]
recall_cyberbullying = [96, 100, 100]
recall_non_cyberbullying = [22, 13, 2]
f1_score_cyberbullying = [77, 92, 89]
f1_score_non_cyberbullying = [34, 29, 4]
accuracy = [66, 86, 80] # Tambahkan nilai akurasi di
sini

# Plotting
plt.figure(figsize=(10, 6))

plt.plot(scenarios, precision_cyberbullying,
marker='o', label='Presisi Cyberbullying')
plt.plot(scenarios, precision_non_cyberbullying,
marker='o', label='Presisi Non-Cyberbullying')
plt.plot(scenarios, recall_cyberbullying, marker='o',
label='Recall Cyberbullying')
plt.plot(scenarios, recall_non_cyberbullying,
marker='o', label='Recall Non-Cyberbullying')
plt.plot(scenarios, f1_score_cyberbullying,
marker='o', label='Skor F1 Cyberbullying')
plt.plot(scenarios, f1_score_non_cyberbullying,
marker='o', label='Skor F1 Non-Cyberbullying')
plt.plot(scenarios, accuracy, marker='o',
label='Akurasi') # Tambahkan akurasi di sini

plt.title('Metrik Kinerja untuk Berbagai Skenario')
plt.xlabel('Skenario')
plt.ylabel('Persentase')
plt.legend()
plt.grid(True)
plt.show()
```