




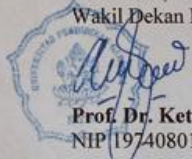
**LAMPIRAN**

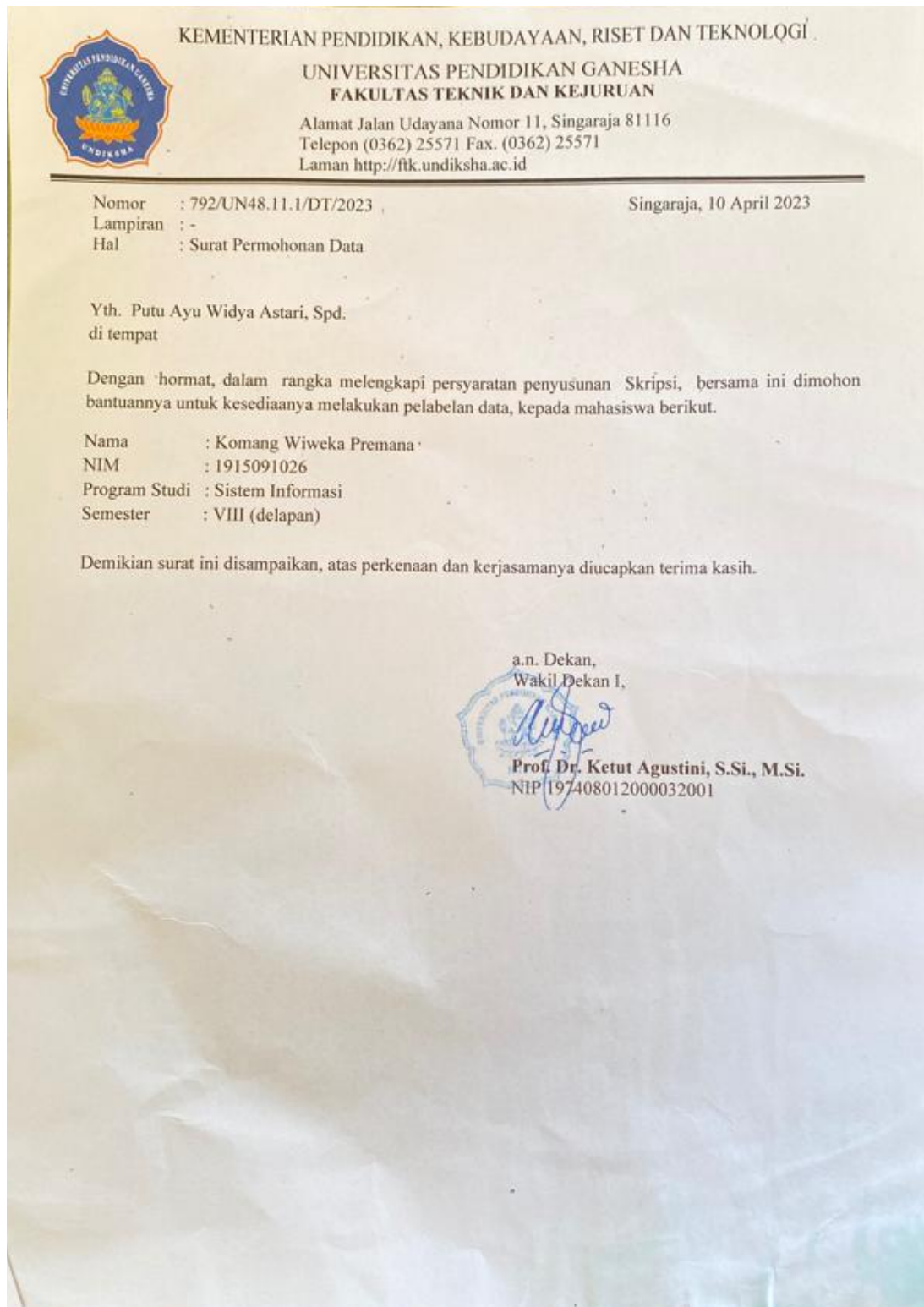
**Lampiran 1. Riwayat Hidup****RIWAYAT HIDUP**

Saya, Komang Wiweka Premana, lahir di Denpasar pada tanggal 2 Oktober 2001. Saya bangga menjadi seorang Warga Negara Indonesia (WNI) dan menjalankan agama Hindu. Saat ini, saya tinggal di Jalan Gg Balangan 1, Ungasan, Kecamatan Kuta Selatan, Kabupaten Badung, Bali. Perjalanan pendidikan saya dimulai di SD N 4 Ungasan, di mana saya berhasil menyelesaikan pendidikan dasar saya dan lulus pada tahun 2013. Setelah itu, saya melanjutkan ke SMP N 2 Kuta Selatan dan meraih kelulusan pada tahun 2016. Setelah menamatkan jenjang SMP, saya memilih melanjutkan pendidikan di SMA N 1 Kuta Selatan dengan mengambil jurusan MIPA. Pada tahun 2019, saya menyelesaikan pendidikan menengah atas dan melanjutkan perjalanan akademik saya di Universitas Pendidikan Ganesha. Saya merasa bangga dapat diterima di universitas tersebut dan memilih program studi Sistem Informasi, Jurusan Teknik Informatika, Fakultas Teknik dan Kejuruan.



**Lampiran 2. Surat Ketersediaan Pelabelan Data**

	<b>KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI</b> <b>UNIVERSITAS PENDIDIKAN GANESHA</b> <b>FAKULTAS TEKNIK DAN KEJURUAN</b> Alamat Jalan Udayana Nomor 11, Singaraja 81116 Telepon (0362) 25571 Fax. (0362) 25571 Laman <a href="http://fbk.undiksha.ac.id">http://fbk.undiksha.ac.id</a>
Nomor : 792/UN48.11.1/DT/2023	Singaraja, 10 April 2023
Lampiran : -	
Hal : Surat Permohonan Data	
 Yth. Putu Ayu Widya Astari, Spd. di tempat	
Dengan hormat, dalam rangka melengkapi persyaratan penyusunan Skripsi, bersama ini dimohon bantuannya untuk kesediaanya melakukan pelabelan data, kepada mahasiswa berikut.	
Nama : Komang Wiweka Premana	
NIM : 1915091026	
Program Studi : Sistem Informasi	
Semester : VIII (delapan)	
Demikian surat ini disampaikan, atas perkenaan dan kerjasamanya diucapkan terima kasih.	
 a.n. Dekan, Wakil Dekan I,  <b>Prof. Dr. Ketut Agustini, S.Si., M.Si.</b> NIP 197408012000032001	



**Lampiran 3. Sertifikat Pendidik**

No. 0000875



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
REPUBLIK INDONESIA

## SERTIFIKAT PENDIDIK

Nomor: 0010481915621007

Berdasarkan Surat Keputusan Menteri Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia Nomor 746/KPT/1/2018 tanggal 05 September 2018 tentang Ijin Pembukaan Program Studi Pendidikan Profesi Guru, Rektor Universitas Pendidikan Ganesha menyatakan bahwa:

**PUTU AYU WIDYA ASTARI**  
Nomor Induk Mahasiswa: 1974805031

lahir di Busungbiu pada tanggal enam bulan Juli tahun seribu sembilan ratus sembilan puluh tiga telah memenuhi semua syarat penyelesaian Pendidikan Profesi Guru dan LULUS Uji Kompetensi Mahasiswa Pendidikan Profesi Guru. kepadanya diberikan sebutan profesi GURU (Gr.) Bahasa Indonesia sesuai hak dan kewajiban yang melekat pada sebutan profesi tersebut.



Singaraja, 18 Desember 2019  
Rektor,

Prof. Dr. I Nyoman Jampel, M.Pd.  
NIP. 195910101986031003



001048192201146103854001481915621007-1577403892

No. 01534



DEPARTEMEN PENDIDIKAN NASIONAL  
REPUBLIK INDONESIA

### SERTIFIKAT PENDIDIK

Nomor : 210915601534

Berdasarkan Surat Keputusan Menteri Pendidikan Nasional Nomor 022/P/2009 tentang Penetapan Perguruan Tinggi Penyelenggara Sertifikasi bagi Guru dalam Jabatan, Rektor Universitas Pendidikan Ganesha, selaku Ketua Rayon 21, menyatakan bahwa :

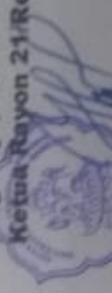
**Dr. ROMANUS ARI SETYAWATI**

Nomor peserta 092206156101534 dinyatakan lulus dan berhak pada tanggal 6 Desember 1960

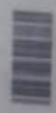
**LULUS Sertifikasi Guru dalam Jabatan dan dinyatakan sebagai GURU PROFESIONAL**  
bidang studi **BAHASA INDONESIA (DAN SASTRA INDONESIA)**



Singaraja, 21 Desember 2009  
Ketua Rayon 21/Rektor,



**Prof. Dr. J Nyoman Sudiana, M.Pd.**  
NIP. 131477325



## Lampiran 4. Surat Keterangan Validasi



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS PENDIDIKAN GANESHA  
FAKULTAS TEKNIK DAN KEJURUAN  
JURUSAN TEKNIK INFORMATIKA  
Alamat Jalan Udayana Nomor 11, Singaraja 81116  
Telepon (0362) 25571 Fax. (0362) 25571  
Laman <http://ftk.undiksha.ac.id>

### SURAT KETERANGAN VALIDASI LABELING DATASET

Yang bertanda tangan di bawah ini.

Nama : Dra. Komang Ari Setyawati

NIP : 196012061987032008

Menerangkan bahwa Mahasiswa Universitas Pendidikan Ganesha di bawah ini:

Nama : Komang Wiweka Premana

NIM : 1915091026

Prodi/Jurusan : Sistem Informasi/ Teknik Informatika

Memang benar bahwa dataset yang sudah dilabeli telah divalidasi pada 14 Juli 2023.  
Demikian surat keterangan ini dibuat dengan sebenarnya untuk dapat digunakan sebagaimana mestinya.

Singaraja, 14 Juli 2023

Ahli/Pakar I,

Dra. Komang Ari Setyawati  
NIP. 196012061987032008



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI**  
**UNIVERSITAS PENDIDIKAN GANESHA**  
FAKULTAS TEKNIK DAN KEJURUAN  
JURUSAN TEKNIK INFORMATIKA  
Alamat Jalan Udayana Nomor 11, Singaraja 81116  
Telepon (0362) 25571 Fax. (0362) 25571  
Laman <http://ftk.undiksha.ac.id>

---

**SURAT KETERANGAN VALIDASI  
LABELING DATASET**

Yang bertanda tangan di bawah ini.

Nama : Putu Ayu Widya Astari, S.Pd.

NIP : 199307062019032015

Menerangkan bahwa Mahasiswa Universitas Pendidikan Ganesha di bawah ini:

Nama : Komang Wiweka Premana

NIM : 1915091026

Prodi/Jurusan : Sistem Informasi/ Teknik Informatika

Memang benar bahwa dataset yang sudah dilabeli telah divalidasi pada 14 Juli 2023.  
Demikian surat keterangan ini dibuat dengan sebenarnya untuk dapat digunakan sebagaimana mestinya.

Singaraja, 14 Juli 2023

Ahli/Pakar II,

Putu Ayu Widya Astari, S.Pd  
NIP. 199307062019032015



### Lampiran 5. Proses *Crawling Data*

```

import snsrape.modules.twitter as sntwitter
import pandas as pd

# Daftar query yang akan digunakan
queries = ['datingapp', 'tinder', 'tantan', 'okcupid',
'kencan online', 'dating online']

# Rentang waktu pencarian tweet
start_date = "2020-12-01"
end_date = "2022-01-01"
# Inisialisasi list untuk menyimpan hasil tweet
tweets = []

# Melakukan crawling untuk setiap query
for query in queries:
    # Menentukan jumlah tweet yang akan diambil per query
    max_tweets = 500

    # Membuat query string dengan bahasa Indonesia
    query_string = f'{query} lang:id'

    # Melakukan crawling menggunakan snsrape
    for i, tweet in
enumerate(sntwitter.TwitterSearchScrapper(query_string).get
_items()):
        tweets.append(tweet.content)

        # Mengecek apakah jumlah tweet yang diambil sudah
mencapai target per query
        if i + 1 == max_tweets:
            break

# Membuat DataFrame dari hasil tweet
tweets_df = pd.DataFrame(tweets, columns=['tanggal',
'username', 'text'])

# Menyimpan DataFrame ke file CSV
df_tweets.to_csv('/Users/wiwepakremana/Documents/Skripsi/j
upyter notebook/data/all_tweets.csv', index=False)

```

### Lampiran 6. Proses *Cleaning Data*

```
# Cleaning data
def clean_text(text):
    # Menghapus karakter non-alphanumeric dan username
    text = re.sub('[^0-9a-zA-Z@]+', ' ', text)
    text = re.sub(r'@\w+', '', text)
    text = re.sub(r'@', '', text)
    # Hapus URL
    text = re.sub(r'http\S+|www\S+|https\S+', '', text)
    # Menghapus angka dalam kata
    text = re.sub(r'\d', '', text)
    # Menghapus spasi di awal dan akhir teks
    text = text.strip()
    # Menghapus duplikasi karakter berlebihan
    text = re.sub(r'(\.)\1+', r'\1', text)
    return text
#mengaplikasikan cleaning data pada setiap tweet
df['text'] = df['text'].apply(clean_text)

# Menampilkan data setelah dilakukan preprocessing
display(df.head(10))
```

### Lampiran 7. Proses *Case Folding*

```
# Fungsi untuk melakukan case folding
def case_folding(text):
    return text.lower()
#mengaplikasikan Case Folding pada setiap tweet
df['text'] = df['text'].apply(case_folding)
# Menampilkan data setelah dilakukan preprocessing
display(df.head(10))
```

### Lampiran 8. Proses Stopword Removal

```

# Fungsi untuk melakukan stopwords removal
def remove_stopwords(text, stopwords):
    words = nltk.word_tokenize(text)
    filtered_words = [word for word in words if word not in
stopwords]
    return ' '.join(filtered_words)

# Baca file eksternal yang berisi stop words tambahan
def read_additional_stopwords(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        additional_stopwords = file.read().splitlines()
    return additional_stopwords

# Baca file eksternal yang berisi stop words tambahan
additional_stopwords =
read_additional_stopwords('/Users/wiwekapremana/Documents/
Skripsi/jupyter notebook/resource/stopword.txt')

# Stop words tambahan manual
manual_stopwords = ['sih', 'trs', 'nya', 'yg',
'kl', 'dah', 'dehh', 'dehhh', 'ngga', 'w', 'aj', 'ga', 'mulu',
'gitu', 'askrl', 'tau', 'gatau', 'mah', 'ih', 'si', 'tuh',
'n',
'tt', 'mana', 'tuh', 'pak', 'ac', 'di', 'chewyena', 'hehe', 'eeh',
'o', 'lah', 'ah', 'u', 'ni', 'loh', 'tu', 'be', 'masi',
'kali', 'nah', 'niii', 'gasi', 'hi', 'nih', 'an',
'aejzcahkl', 'lu',
'gimana', 'gaje', 'yah', 'kan', 'pas', 'say', 'loh', 'nga', 'deh',
'dech', 'jah', 'lo', 'mangkanya', 'kok', 'gimana', 'alo',
'gk',
'ga', 'gw', 'gue', 'guee', 'ku', 'hah', 'g', 'gak', 'apa', 'ap', 'pa',
'p', 'sdh', 'dh', 'udah', 'bgt', 'y', 'persen', 'ala']

# Menggabungkan stop words bawaan dengan stop words tambahan
stop_words = set(stopwords.words('indonesian') +
additional_stopwords + manual_stopwords)

```

### Lampiran 9. Proses Tokenizing

```
# Fungsi untuk melakukan tokenisasi
def tokenize(text):
    tokens = nltk.word_tokenize(text)
    return tokens

#mengaplikasikan Tokenizing pada setiap tweet
df['text'] = df['text'].apply(tokenize)

# Menampilkan data setelah dilakukan preprocessing
display(df.head(10))
```

### Lampiran 10. Proses Stemming

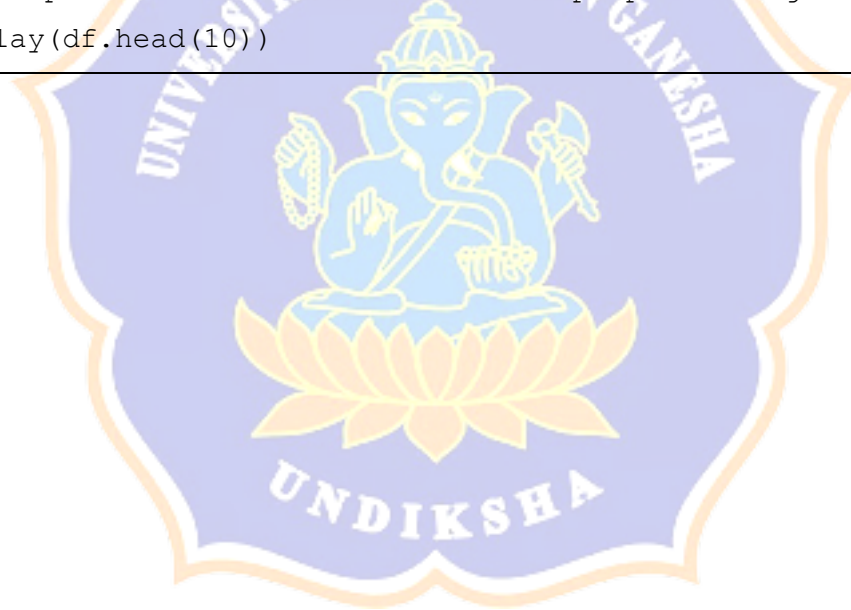
```
# Fungsi untuk melakukan stemming
def stem(tokens):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    stemmed_tokens = [stemmer.stem(token) for token in
tokens]
    return stemmed_tokens

# Membaca file eksternal
with open('/Users/wiwepakremana/Documents/Skripsi/jupyter
notebook/resource/kata-dasar.txt', 'r') as file:
    content = file.read()
    tokens = content.split() # Mengambil token-token dari
file

#mengaplikasikan cleaning data pada setiap tweet
df['text'] = df['text'].apply(lambda x: stem(x))
display(df.head(10))
```

### Lampiran 11. Proses Normalisasi

```
# proses normalisasi
normalized_word = pd.read_csv(
    "/Users/wiwekapremana/Documents/Skripsi/jupyter
notebook/resource/normalisasi.csv")
normalized_word_dict = {}
for index, row in normalized_word.iterrows():
    if row[0] not in normalized_word_dict:
        normalized_word_dict[row[0]] = row[1]
def normalized_term(document):
    return [normalized_word_dict[term] if term in
normalized_word_dict else term for term in document]
df['text'] = df['text'].apply(normalized_term)
# Menampilkan data setelah dilakukan preprocessing
display(df.head(10))
```



### Lampiran 12. Proses Seleksi Fitur *Information Gain*

```

# Membuat CountVectorizer untuk mengubah teks menjadi
matriks token count
vectorizer = CountVectorizer()
X_vectorized = vectorizer.fit_transform(X['text'])
# Melakukan seleksi fitur menggunakan informasi gain
ig = 900 # Pilih jumlah fitur terbaik
selector = SelectKBest(score_func=mutual_info_classif,
k=ig)
X_selected = selector.fit_transform(X_vectorized, y)
# Menampilkan fitur-fitur yang dipilih setelah seleksi
dengan Information Gain
selected_feature_indices =
selector.get_support(indices=True)
selected_features =
vectorizer.get_feature_names_out()[selected_feature_indices]
# Mengambil skor Information Gain dari selector
info_gain_scores = selector.scores_
# Mengambil nama fitur dari vectorizer
feature_names = vectorizer.get_feature_names_out()
# Membuat dataframe untuk menampilkan skor Information Gain
info_gain_df = pd.DataFrame({'Feature': feature_names,
'Information Gain': info_gain_scores})
# Mengurutkan dataframe berdasarkan skor Information Gain
secara menurun
info_gain_df = info_gain_df.sort_values(by='Information
Gain', ascending=False)
# Mendapatkan jumlah fitur yang tersedia
jumlah_fitur = X_vectorized.shape[1]
print(selected_features)
print()

```

### Lampiran 13. Proses Klasifikasi

```

# Melakukan k-fold cross-validation
kfold = KFold(n_splits=10, shuffle=True, random_state=42)
# Menginisialisasi list untuk menyimpan confusion matrix
dari setiap fold
confusion_matrices = []
# Menginisialisasi list untuk menyimpan classification
report dari setiap fold
classification_reports = []
# Melakukan iterasi pada setiap fold
for train_index, test_index in kfold.split(X_selected):
    # Memisahkan data menjadi set training dan uji
    X_train, X_test = X_selected[train_index],
X_selected[test_index]
    y_train, y_test = y[train_index], y[test_index]
    # Mendefinisikan grid parameter untuk grid search
    param_grid = {'alpha': [0.1, 1.0, 10.0]}
    # Melakukan grid search dengan cross-validation
    classifier = GridSearchCV(MultinomialNB(), param_grid,
cv=10)
    classifier.fit(X_train, y_train)
    # Melakukan prediksi pada set uji
    y_pred = classifier.predict(X_test)
    # Membuat classification report
    classification = classification_report(y_test, y_pred,
output_dict=True)
    classification_reports.append(pd.DataFrame(classification)
.transpose())
    # Membuat confusion matrix
    confusion = confusion_matrix(y_test, y_pred)
    confusion_matrices.append(confusion)
    print('Confusion Matrix:')
    print(confusion)
    print()
# Menghitung rata-rata confusion matrix

```

```

avg_confusion_matrix = np.mean(confusion_matrices, axis=0)
avg_confusion_matrix =
np.round(avg_confusion_matrix).astype(int)
labels = classifier.classes_
print('Hasil Klasifikasi Naive Bayes dan Information Gain')
print('Average Confusion Matrix:')
print(avg_confusion_matrix)
print()
# Menghitung rata-rata classification report
avg_classification_report =
pd.concat(classification_reports).groupby(level=0).mean().
round(2)
print('Average Classification Report:')
print(avg_classification_report)

```

#### Lampiran 14. Proses Pengujian Data Baru

```

new_data = pd.DataFrame({'text': ['main dating app
bagus banget, terbaik lah pokoknya','temen gue pada
ngomongin kebusukan ditinder tiap kumpul ','dating app
punya fitur yang lengkap, keren','Saya merasa tidak
nyaman dengan aplikasi dating online ini karena pernah
kena tipu','main tinder aja harus bayar, males
banget']})
# Mengubah teks baru menjadi matriks token count
new_data_vectorized =
vectorizer.transform(new_data['tex
# Mengaplikasikan seleksi fitur yang sama pada data
baru
new_data_selected =
selector.transform(new_data_vectorized)
# Melakukan prediksi pada data baru
predictions = classifier.predict(new_data_selected)
# Menampilkan hasil prediksi
for text, prediction in zip(new_data['text'],
predictions):
    print(f>Data: {text}")
    print(f>Prediksi: {prediction}")
    print()

```