Lampiran 1. Lampiran 1 Data Mahasiswa Pelabel Data

DATA MAHASISWA PELABEL DATA

Mahasiswa 1

Nama	Nurul Humairah
Prodi	Pendidikan Bahasa Indonesia
Semester	8
IPK	3,62

Mahasiswa 2

Nama	Sita Adelia
Prodi	Pendidikan Bahasa Indonesia
Semester	8
IPK	3,74

Mahasiswa 3

Nama	Satrio Bagus Alvian
Prodi	Pendidikan Bahasa Indonesia
Semester	8
IPK	3,79

Lampiran 2. *Source Code Program*

```
# clean Text

def cleanText(title):

    title = re.sub(r'[0-9]+', '', title) # hapus bilangan angka

    title = re.sub('"', '', title) # menghilangkan petik

    title = title.replace('\n', ' ') # mengganti baris baru ke
dalam space

    # title = title.replace('-', ' ')

    title = re.sub(r'(\w+)-\1', r'\1', title) # kata-kata duplikat

    title = title.translate(str.maketrans('', '',
string.punctuation)) # hapus semua tanda baca

    title = title.strip(' ') # hapus spasi karakter dari teks kiri
dan kanan

    title = re.sub(r'\b\w{1,3}\b','',title) #hapus kata kurang
dari 3

    return title

data['title_clean'] = data['Title'].apply(cleanText)

data
```

L1. Code *Cleaning* Data

```
# lower case

def casefoldingText(title): # Mengubah semua karakter dalam teks
menjadi huruf kecil

   title = title.lower()

   return title

data['title_casfolding'] =
data['title_clean'].apply(casefoldingText)

data
```

L2. Code *Case Folding*

```
# Tokenizine
def tokenizingText(title):
  title= word_tokenize(title)
  return title
data['title_tokenize'] = data
['title_casfolding'].apply(tokenizingText)
data
```

L3. Code *Tokenizing* Data

```
def filteringText(title): # deklarasi fungsi
    listStopwords = set(stopwords.words('indonesian'))
    #list menyimpan kata di luar stopword
    filtered = []
    # loop for untuk setiap teks dalam title.
    for txt in title:
        if txt not in listStopwords:
            filtered.append(txt)
    title = filtered
    return title
data['title_stopword'] = data
['title_tokenize'].apply(filteringText)
data
```

L4. Code *Stopword* Data

```
def stemmingText(title): # deklarasi fungsi
    # Membuat objek berita
    berita = StemmerFactory()
    stemmer = berita.create_stemmer()
    # list untuk menyimpan hasil stemming
    hasil_stemming = []
    title = [stemmer.stem(word) for word in title]
    hasil_stemming = title
    return title
data['title_stemming'] = data
['title_stopword'].apply(stemmingText)
data
```

L4. *Code Stemming Data*

```
import pickle
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(ngram_range=(1,1), binary=True,
max_features=3000)
features = vectorizer.fit_transform(data['sentences'])
with open('vectorizer.pkl', 'wb') as f:
    pickle.dump(vectorizer, f)
count_feature_names =  vectorizer.get_feature_names_out()
count_vector = pd.DataFrame(
    features.todense(),
              columns= vectorizer.get_feature_names_out()
              )
count_vector
```

L5. Code ekstraksi fitur

```
from sklearn.model_selection import KFold
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
import pickle
# Split dataset into X (text) and y (sentiment)
X = data1  # title
y = data2  # label
# Define k-fold cross-validation
kf = KFold(n_splits=25, random_state=25, shuffle=True)
# Initialize lists to store evaluation metrics
accuracy_scores = []
confusion_matrices = []
# Iterate through each fold
for train_index, val_index in kf.split(X):
    # Split data into training and testing sets for current fold
    X_train, X_val = X.iloc[train_index], X.iloc[val_index]
    y_train, y_val = y.iloc[train_index], y.iloc[val_index]
```

```
    # Train the SVM model
    model_svm = svm.SVC(kernel='rbf', C=100, gamma='scale',
class_weight='balanced')
    #gamma=scale
    model_svm.fit(X_train, y_train)
    #, class_weight='balanced'
    # Evaluate the model on the testing set
    y_pred = model_svm.predict(X_val)
    accuracy_scores.append(accuracy_score(y_val, y_pred))
    # Compute the confusion matrix for the current fold
    cm = confusion_matrix(y_val, y_pred)
    confusion_matrices.append(cm)
# Store the accuracy results in a dataframe
data = pd.DataFrame({'accuracy': accuracy_scores})
print(data)
# Find the best model based on accuracy
best_model_index = data['accuracy'].idxmax()
best_model = model_svm
# Save the best model to a file
filename = 'best_model20.pkl'
with open(filename, 'wb') as file:
    pickle.dump(best_model, file)
```

L6. Code Model SVM

```
import pickle
# memuat model vectorizer yang telah disimpan
with open('vectorizer.pkl', 'rb') as f:
    vectorizer = pickle.load(f)
# memuat model klasifikasi svm yang telah disimpan
with open('best_model20.pkl', 'rb') as f:
    model_svm = pickle.load(f)
 # membaca data aktual dari file CSV
actual_data = pd.read_csv('/content/Label  Pengujian 370 - 1.csv')
```

```
new_countvectorizer = vectorizer.transform(data['title_sentence'])

# convert the sparse matrix to a dense matrix

X_test = new_countvectorizer.toarray()

# membuat prediksi pada data uji menggunakan model SVM

pred_new = model_svm.predict(X_test)

# menambahkan kolom sentimen prediksi (sentiment_pred) pada
dataframe data

data['pred'] = pred_new

# menambahkan kolom sentimen aktual (sentiment_actual) pada
dataframe data

data['actual'] = actual_data

# print hasil

print(data[['title', 'actual', 'pred']])

# Menyimpan data ke dalam file CSV

data[['title', 'actual', 'pred']].to_csv('hasil_prediksi2.csv',
index=False)
```

L7. Code Testing Data

RIWAYAT HIDUP

Ni Luh Putu Risma Dewi lahir di Timika pada tanggal 7 Desember. Penulis lahir dari pasangan suami istri Bapak I Made Sujana dan ibu Rosalina Sainah. Penulis berkebangsaan Indonesia dan beragama Hindu. Kini penulis beralamat di Asrama Kodim Kubujati, No.18, Kecamatan Buleleng, Kabupaten Buleleng, Provinsi Bali. Penulis menyelesaikan pendidikan dasar di SD Inpres Timika 3 dan lulus pada 2013. Kemudian penulis melanjutkan di SMP Negeri 2 Mimika dan lulus pada tahun 2016. Pada tahun 2019, penulis lulus dari SMA Negeri 3 Singaraja jurusan MIPA. Penulis terdaftar sebagai mahasiswa Program Studi S1 Ilmu Komputer di Universitas Pendidikan Ganesha.


UNIVERSITAS PENDIDIKAN GANESHA
UNDIKSHA