



LAMPIRAN

Lampiran 1. Identitas mahasiswa sebagai pembantu peneliti.

DATA MAHASISWA PELABEL DATA

	Nama	Program Studi	Semester	IPK
1	Nurul Humairah	Pendidikan Bahasa dan Sastra Indonesia	8	3.62
2	Nyoman Rai Suwija	Pendidikan Bahasa dan Sastra Indonesia	6	3.80
3	Verunia Kamalini	Pendidikan Bahasa dan Sastra Indonesia	6	3.57
4	Milda Putri Rizqiwati	Pendidikan Bahasa dan Sastra Indonesia	6	3.70

Lampiran 2. *Source code program*

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_selection import SelectKBest, chi2
import pickle

# inialisasi fungsi countvector untuk mengubah teks menjadi
matriks
vectorizer = CountVectorizer(ngram_range=(1,1),binary=True)
X = vectorizer.fit_transform(data['sentences'])

# inialisasi fungsi selectKbest dengan seleksi chi2
ch2 = SelectKBest(chi2, k=2100)
X_selected = ch2.fit_transform(X,data["label"])

# Get the selected feature names
feature_names = vectorizer.get_feature_names_out()
selected_feature_indices = ch2.get_support(indices=True)
selected_feature_names = [feature_names[i] for i in
selected_feature_indices]
```

```

New_X = pd.DataFrame(X_selected.toarray(),
columns=selected_feature_names)
New_X
with open('ch2.pkl', 'wb') as f:
    pickle.dump(ch2, f)

```

L.1 Code proses seleksi fitur *Chi-Square*

```

import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_selection import SelectKBest,
mutual_info_classif
import pickle
# inialisasi fungsi countvector untuk mengubah teks menjadi
matriks
vectorizer = CountVectorizer(ngram_range=(1,1),binary=True)
X = vectorizer.fit_transform(data['sentences'])

# inialisasi fungsi selectKbest dengan seleksi information gain
ig = SelectKBest(mutual_info_classif, k=2200)
X_selected = ig.fit_transform(X,data["label"])

# Get the selected feature names
feature_names = vectorizer.get_feature_names_out()
selected_feature_indices = ig.get_support(indices=True)
selected_feature_names = [feature_names[i] for i in
selected_feature_indices]

New_X = pd.DataFrame(X_selected.toarray(),
columns=selected_feature_names)
New_X
with open('ig.pkl', 'wb') as f:
    pickle.dump(ig, f)

```

L.2 Code proses seleksi fitur *Information Gain*.

```

from sklearn.svm import SVC
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import KFold
# Split dataset into X (sentimnet) and y (label)

```

```

# X = New_X #title
X = data1 #title
y = data2 #label
# Define k-fold cross-validation
kf = KFold(n_splits=35, random_state=10, shuffle=True)
#list menyimpan evaluasi matrix
accuracy_scores = []
confusion_matrices = []
# Iterate through each fold
for train_index, test_index in kf.split(X):
    # Split data into training and testing sets for current fold
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]
    # Train the SVM model
    clf = svm.SVC(kernel='rbf', C=100,
gamma="scale", class_weight='balanced')
    clf.fit(X_train, y_train)
    # Evaluate the model on the testing set
    y_pred = clf.predict(X_test)
    accuracy_scores.append(accuracy_score(y_test, y_pred))
    #Compute the confusion matrix for the current fold
    cm = confusion_matrix(y_test, y_pred)
    confusion_matrices.append(cm)
# Store the accuracy results in a dataframe
data_acc = pd.DataFrame({'accuracy': accuracy_scores})
print(data_acc)

```

L.3 Code training SVM

```

import gensim
from gensim import corpora
dictionary = corpora.Dictionary(doc_clean)
print(dictionary)
doc_term_matrix = [dictionary.doc2bow(doc) for doc in doc_clean]
# Creating the object for LDA model using gensim library
Lda = gensim.models.ldamodel.LdaModel
total_topics = 5 # jumlah topik yang akan di extract
number_words = 10 # jumlah kata per topik
# Running and Trainign LDA model on the document term matrix.
lda_model = Lda(doc_term_matrix, num_topics=total_topics, id2word
= dictionary, passes=50)
lda_model.show_topics(num_topics=total_topics,
num_words=number_words)

```

```

# Word Count of Topic Keywords
from collections import Counter
topics = lda_model.show_topics(formatted=False)
data_flat = [w for w_list in doc_clean for w in w_list]
counter = Counter(data_flat)
out = []
for i, topic in topics:
    for word, weight in topic:
        out.append([word, i , weight, counter[word]])
df_imp_wcount = pd.DataFrame(out, columns=['word', 'topic_id',
'importance', 'word_count'])
# print(df_imp_wcount)
df_imp_wcount

```

L.4 Code Pemodelan Topik LDA

```

import pyLDAvis.gensim
import pickle
import pyLDAvis
# Visualize the topics
pyLDAvis.enable_notebook()
import os
LDAvis_data_filepath =
os.path.join('ldavis_prepared_'+str(total_topics))
corpus = [dictionary.doc2bow(text) for text in doc_clean]
if 1 == 1:
    LDAvis_prepared = pyLDAvis.gensim.prepare(lda_model, corpus,
dictionary)
    with open(LDAvis_data_filepath, 'wb') as f:
        pickle.dump(LDAvis_prepared, f)
with open(LDAvis_data_filepath, 'rb') as f:
    LDAvis_prepared = pickle.load(f)
LDAvis_prepared

```

L.5 Code Visualisasi LDA dengan *Intertopic Distance Map*

RIWAYAT HIDUP



I Putu Soma Darmayasa lahir di Singaraja pada tanggal 2 Oktober 2000. Penulis lahir dari pasangan suami istri Bapak I Made Kertanegara dan Ibu Ni Ketut Puspa. Penulis berkebangsaan Indonesia dan beragama Hindu. Kini penulis beralamat di Jl. Cok Gede Rai No.30 Peliatan, Kecamatan Ubud, Kabupaten Gianyar, Provinsi Bali. Penulis menyelesaikan pendidikan dasar di SDN 4 Peliatan dan lulus pada 2013. Kemudian penulis melanjutkan di SMP Negeri 1 Ubud dan lulus pada tahun 2016. Pada tahun 2019, penulis lulus dari SMAN 1 Gianyar dengan jurusan IPA. Penulis terdaftar sebagai mahasiswa Program Studi S1 Ilmu Komputer di Universitas Pendidikan Ganesha pada tahun 2019.

