

Lampiran 3 Kode Program Metode KNN

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_recall_fscore_support
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
# Memuat dataset latih
df_klasifikasi_train = pd.read_csv("datasettrainingconv.csv")
# Memuat dataset uji
df_klasifikasi_test = pd.read_csv("datasettestingconv.csv")

print(df_klasifikasi_train)

print(df_klasifikasi_test)

X_test_new = df_klasifikasi_test[['NAMA', 'PENDIDIKAN', 'JT',
'PENGHASILAN', 'PENGELUARAN', 'SA', 'SKR', 'DL', 'TRANSPORTASI',
'UMUR']]
y_test_new = df_klasifikasi_test['KATEGORI']
X_train = df_klasifikasi_test[['NAMA', 'PENDIDIKAN', 'JT',
'PENGHASILAN', 'PENGELUARAN', 'SA', 'SKR', 'DL', 'TRANSPORTASI',
'UMUR']]
y_train = df_klasifikasi_test['KATEGORI']
# Definisikan preprocessor untuk fitur numerik dan kategorikal
numerical_features = ['PENDIDIKAN', 'JT', 'PENGHASILAN',
'PENGELUARAN', 'SA', 'SKR', 'DL', 'TRANSPORTASI', 'UMUR']
categorical_features = ['NAMA']
preprocessor = ColumnTransformer(
transformers=[
('num', StandardScaler(), numerical_features),
('cat', OneHotEncoder(), categorical_features)])
# Pipeline: Preprocessor + KNN
pipeline = Pipeline(steps=[('preprocessor', preprocessor),
('classifier', KNeighborsClassifier(n_neighbors=5))])
# Latih model KNN pada dataset latih
pipeline.fit(X_train, y_train)
# Lakukan prediksi pada dataset uji yang baru
y_pred_new = pipeline.predict(X_test_new)
```

```

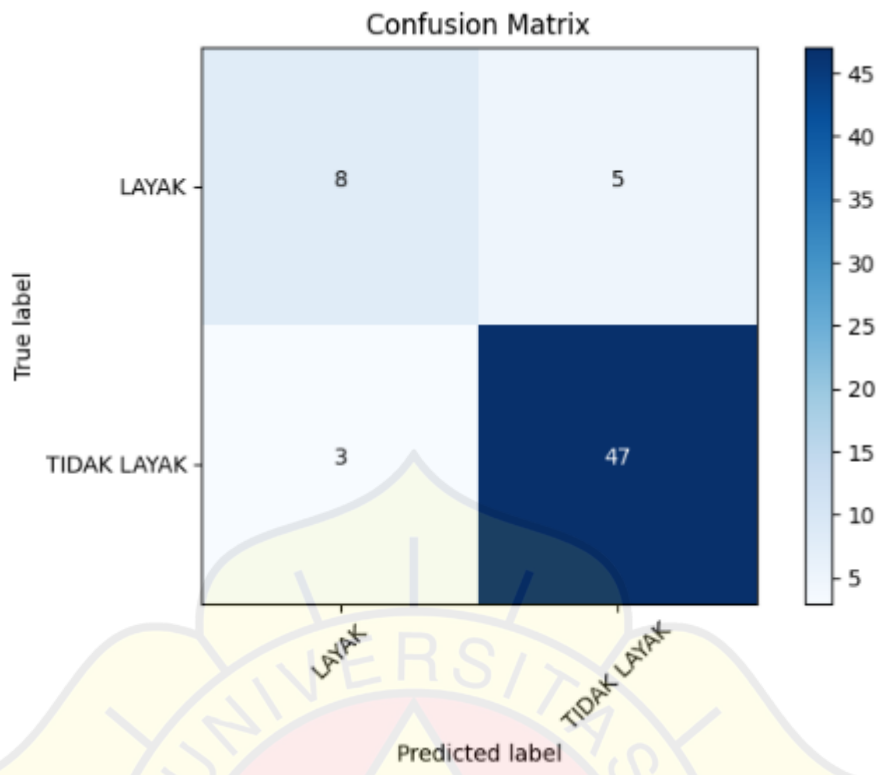
# Hitung matriks kebingungan
cm_new = confusion_matrix(y_test_new, y_pred_new)

plt.figure(figsize=(10, 8))
plt.imshow(cm_new, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()
classes = np.unique(y_test_new)
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45)
plt.yticks(tick_marks, classes)

for i in range(cm_new.shape[0]):
    for j in range(cm_new.shape[1]):
        plt.text(j, i, format(cm_new[i, j], 'd'),
                 horizontalalignment="center",
                 color="white" if cm_new[i, j] > cm_new.max() / 2
else "black")

plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.tight_layout()
plt.show()
# Hitung akurasi pada dataset uji yang baru
accuracy_new = accuracy_score(y_test_new, y_pred_new)
print('Akurasi Testing KNN pada Dataset Baru:', accuracy_new)
# Hitung precision, recall, dan f-score pada dataset uji yang baru
P_new = precision_recall_fscore_support(y_test_new, y_pred_new,
average='macro')
print("Precision : ", P_new[0])
print("Recall : ", P_new[1])
print("F Score : ", P_new[2])

```



Akurasi Testing KNN pada Dataset Baru: 0.873015873015873
 Precision : 0.8155594405594406
 Recall : 0.7776923076923077
 F Score : 0.7941176470588236



Lampiran 4 Kode Program Metode MOORA

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.moora import SVC
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_recall_fscore_support
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
# Memuat dataset latih
df_klasifikasi_train = pd.read_csv("datasettrainingconv.csv")
# Memuat dataset uji
df_klasifikasi_test = pd.read_csv("datasettestingconv.csv")
# Pisahkan fitur dan target untuk dataset latih
X_train = df_klasifikasi_test[['NAMA', 'PENDIDIKAN', 'JT',
'PENGHASILAN', 'PENGELUARAN', 'SA', 'SKR', 'DL', 'TRANSPORTASI',
'UMUR']]
y_train = df_klasifikasi_test['KATEGORI']
# Pisahkan fitur dan target untuk dataset uji
X_test_new = df_klasifikasi_test[['NAMA', 'PENDIDIKAN', 'JT',
'PENGHASILAN', 'PENGELUARAN', 'SA', 'SKR', 'DL', 'TRANSPORTASI',
'UMUR']]
y_test_new = df_klasifikasi_test['KATEGORI']
# Definisikan preprocessor untuk fitur numerik dan kategorikal
numerical_features = ['PENDIDIKAN', 'JT', 'PENGHASILAN',
'PENGELUARAN', 'SA', 'SKR', 'DL', 'TRANSPORTASI', 'UMUR']
categorical_features = ['NAMA']
preprocessor = ColumnTransformer(
transformers=[
('num', StandardScaler(), numerical_features),
('cat', OneHotEncoder(), categorical_features)])
# Pipeline: Preprocessor + MOORA
pipeline = Pipeline(steps=[('preprocessor', preprocessor),
('classifier', SVC(kernel='linear', random_state=42))])
# Latih model MOORA pada dataset latih
pipeline.fit(X_train, y_train)
# Lakukan prediksi pada dataset uji yang baru
y_pred_new = pipeline.predict(X_test_new)
# Hitung matriks kebingungan
cm_new = confusion_matrix(y_test_new, y_pred_new)
# Visualisasikan matriks kebingungan
plt.figure(figsize=(10, 8))
plt.imshow(cm_new, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()
```

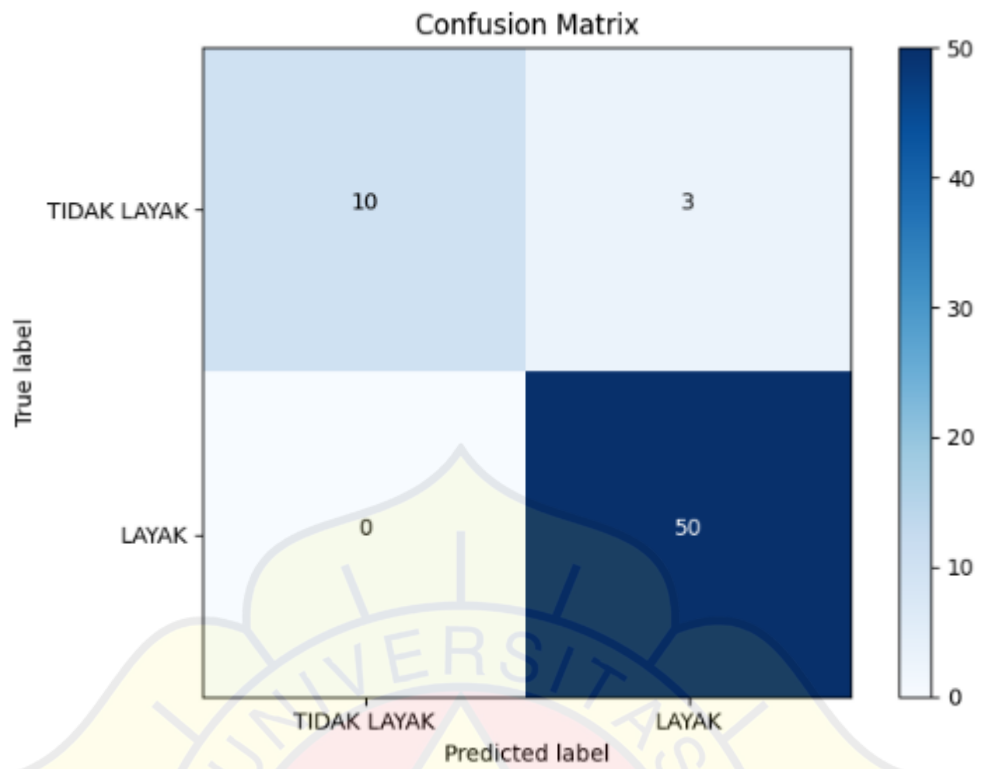
```

classes = y_test_new.unique() # Mengambil kelas unik dari
y_test_new
tick_marks = range(len(classes))
plt.xticks(tick_marks, classes)
plt.yticks(tick_marks, classes)

for i in range(cm_new.shape[0]):
    for j in range(cm_new.shape[1]):
        plt.text(j, i, format(cm_new[i, j], 'd'),
                 horizontalalignment="center",
                 color="white" if cm_new[i, j] > cm_new.max() / 2
else "black")

plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.tight_layout()
plt.show()
# Hitung akurasi pada dataset uji yang baru
accuracy_new = accuracy_score(y_test_new, y_pred_new)
print('Akurasi Testing MOORA pada Dataset Baru:', accuracy_new)
# Hitung precision, recall, dan f-score pada dataset uji yang baru
P_new = precision_recall_fscore_support(y_test_new, y_pred_new,
average='macro')
print("Precision : ", P_new[0])
print("Recall : ", P_new[1])
print("F Score : ", P_new[2])

```



Akurasi Testing MOORA pada Dataset Baru: 0.9523809523809523
 Precision : 0.9716981132075472
 Recall : 0.8846153846153846
 F Score : 0.9202195018995357