

DTW-Single Template

Anggota :

Ronggur Mahendra - 13519008

Rafli Zamzami - 13519067

Raden Haryasatyo Wisjununandono - 1352070

```
In [183...]: !pip install python_speech_features fastdtw
Requirement already satisfied: python_speech_features in /usr/local/lib/python3.10/dist-packages (0.6)
Requirement already satisfied: fastdtw in /usr/local/lib/python3.10/dist-packages (0.3.4)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from fastdtw) (1.23.5)
```

```
In [184...]: import numpy as np
from fastdtw import fastdtw
import math
from scipy.spatial.distance import euclidean, sqeuclidean, cosine, correlation, chebyshev

def compute_cost_matrix(input_array, template):
    distance_matrix = np.zeros((len(template), len(input_array)))
    for i in range(len(distance_matrix)):
        for j in range(len(distance_matrix[0])):
            distance_matrix[i][j] = euclidean(input_array[j], template[i])

    return distance_matrix

def euclidean(a, b):
    total = 0
    for i in range(len(a)):
        total += (a[i]-b[i])**2
    return math.sqrt(total)

def compute_accumulated_cost_matrix(C):
    N = C.shape[0]
    M = C.shape[1]
    D = np.zeros((N, M))
    D[0, 0] = C[0, 0]
    for n in range(1, N):
        D[n, 0] = D[n-1, 0] + C[n, 0]
    for m in range(1, M):
        D[0, m] = D[0, m-1] + C[0, m]
    for n in range(1, N):
        for m in range(1, M):
            D[n, m] = C[n, m] + min(D[n-1, m], D[n, m-1], D[n-1, m-1])
    return D
```

```
In [186...]: from python_speech_features import mfcc
from python_speech_features import logfbank
```

```
import scipy.io.wavfile as wav
import os

folder_path = './dictionary/'
template = []
if os.path.exists(folder_path) and os.path.isdir(folder_path):
    # List all files in the folder
    files = os.listdir(folder_path)

    # Iterate through the files
    for file_name in files:
        # You can process each file here, for example, print the file name
        print(file_name)
        (rate,sig) = wav.read(folder_path + file_name)
        print("signal length and channel : ", sig.shape )
        mfcc_feat = mfcc(sig,rate,nfilt = 39,numcep = 39,nfft =2048)

        print("feature extraction length and channel : ", mfcc_feat.shape)
        template.append([file_name[:-4],mfcc_feat])
```

```
tugas.wav
signal length and channel : (61056, 2)
feature extraction length and channel : (253, 39)
merupakan.wav
signal length and channel : (70272, 2)
feature extraction length and channel : (292, 39)
kuliah.wav
signal length and channel : (81792, 2)
feature extraction length and channel : (340, 39)
kecil.wav
signal length and channel : (56448, 2)
feature extraction length and channel : (234, 39)
ini.wav
signal length and channel : (61056, 2)
feature extraction length and channel : (253, 39)
dictionary.wav
signal length and channel : (87552, 2)
feature extraction length and channel : (364, 39)
halo.wav
signal length and channel : (56448, 2)
feature extraction length and channel : (234, 39)
tambahan.wav
signal length and channel : (73728, 2)
feature extraction length and channel : (306, 39)
suara.wav
signal length and channel : (70272, 2)
feature extraction length and channel : (292, 39)
pemrosesan.wav
signal length and channel : (72576, 2)
feature extraction length and channel : (301, 39)
.ipynb_checkpoints
```

```

-----
IsADirectoryError                                Traceback (most recent call last)
<ipython-input-186-80be9ea2aaa3> in <cell line: 8>()
      14         # You can process each file here, for example, print the file name
      15         print(file_name)
---> 16         (rate,sig) = wav.read(folder_path + file_name)
      17         print("signal length and channel : ", sig.shape )
      18         mfcc_feat = mfcc(sig,rate,nfilt = 39,numcep = 39,nfft =2048)

/usr/local/lib/python3.10/dist-packages/scipy/io/wavfile.py in read(filename, mmap)
 645         mmap = False
 646     else:
--> 647         fid = open(filename, 'rb')
 648
 649     try:
IsADirectoryError: [Errno 21] Is a directory: './dictionary/.ipynb_checkpoints'

```

import suara rafli sebagai dataset

```

In [187...]
folder_path = './test/rafli/'
test_rafli = []
if os.path.exists(folder_path) and os.path.isdir(folder_path):
    # List all files in the folder
    files = os.listdir(folder_path)

    # Iterate through the files
    for file_name in files:
        # You can process each file here, for example, print the file name
        print(file_name)
        (rate,sig) = wav.read(folder_path + file_name)
        print("signal length and channel : ", sig.shape )
        mfcc_feat = mfcc(sig,rate,nfilt = 39,numcep = 39,nfft =2048)

        print("feature extraction length and channel : ", mfcc_feat.shape)
        test_rafli.append([file_name[:-4],mfcc_feat])

```

```
tugas.wav
signal length and channel : (67584,)
feature extraction length and channel : (140, 39)
merupakan.wav
signal length and channel : (81920,)
feature extraction length and channel : (170, 39)
kuliah.wav
signal length and channel : (73728,)
feature extraction length and channel : (153, 39)
kecil.wav
signal length and channel : (73728,)
feature extraction length and channel : (153, 39)
ini.wav
signal length and channel : (73728,)
feature extraction length and channel : (153, 39)
dictionary.wav
signal length and channel : (66560,)
feature extraction length and channel : (138, 39)
halo.wav
signal length and channel : (71680,)
feature extraction length and channel : (148, 39)
tambahan.wav
signal length and channel : (70656,)
feature extraction length and channel : (146, 39)
suara.wav
signal length and channel : (60416,)
feature extraction length and channel : (125, 39)
pemrosesan.wav
signal length and channel : (82944,)
feature extraction length and channel : (172, 39)
```

In [188...]: len(test_rafli)

Out[188]: 10

import suara ronggur sebagai dataset

```
In [189...]: folder_path = './test/ronggur/'
test_ronggur = []
if os.path.exists(folder_path) and os.path.isdir(folder_path):
    # List all files in the folder
    files = os.listdir(folder_path)

    # Iterate through the files
    for file_name in files:
        # You can process each file here, for example, print the file name
        print(file_name)
        (rate,sig) = wav.read(folder_path + file_name)
        print("signal length and channel : ", sig.shape )
        mfcc_feat = mfcc(sig,rate,nfilt = 39,numcep = 39,nfft = 2048)

        print("feature extraction length and channel : ", mfcc_feat.shape)
        test_ronggur.append([file_name[:-4],mfcc_feat])
```

```
tugas.wav
signal length and channel : (73728, 2)
feature extraction length and channel : (306, 39)
merupakan.wav
signal length and channel : (94208, 2)
feature extraction length and channel : (392, 39)
kuliah.wav
signal length and channel : (82944, 2)
feature extraction length and channel : (345, 39)
kecil.wav
signal length and channel : (93184, 2)
feature extraction length and channel : (387, 39)
ini.wav
signal length and channel : (90112, 2)
feature extraction length and channel : (374, 39)
dictionary.wav
signal length and channel : (96256, 2)
feature extraction length and channel : (400, 39)
halo.wav
signal length and channel : (140288, 2)
feature extraction length and channel : (584, 39)
tambahan.wav
signal length and channel : (97280, 2)
feature extraction length and channel : (404, 39)
suara.wav
signal length and channel : (87040, 2)
feature extraction length and channel : (362, 39)
pemrosesan.wav
signal length and channel : (93184, 2)
feature extraction length and channel : (387, 39)
```

In [190...]: len(test_ronggur)

Out[190]: 10

```
In [191...]: folder_path = './test/nando/'
test_nando = []
if os.path.exists(folder_path) and os.path.isdir(folder_path):
    # List all files in the folder
    files = os.listdir(folder_path)

    # Iterate through the files
    for file_name in files:
        # You can process each file here, for example, print the file name
        print(file_name)
        (rate,sig) = wav.read(folder_path + file_name)
        print("signal length and channel : ", sig.shape )
        mfcc_feat = mfcc(sig,rate,nfilt = 39,numcep = 39,nfft =2048)

        print("feature extraction length and channel : ", mfcc_feat.shape)
        test_nando.append([file_name[:-4],mfcc_feat])
```

```
tugas.wav
signal length and channel : (47232, 2)
feature extraction length and channel : (196, 39)
merupakan.wav
signal length and channel : (66816, 2)
feature extraction length and channel : (277, 39)
kuliah.wav
signal length and channel : (47232, 2)
feature extraction length and channel : (196, 39)
kecil.wav
signal length and channel : (87552, 2)
feature extraction length and channel : (364, 39)
ini.wav
signal length and channel : (61056, 2)
feature extraction length and channel : (253, 39)
dictionary.wav
signal length and channel : (61056, 2)
feature extraction length and channel : (253, 39)
halo.wav
signal length and channel : (67968, 2)
feature extraction length and channel : (282, 39)
tambahan.wav
signal length and channel : (44928, 2)
feature extraction length and channel : (186, 39)
suara.wav
signal length and channel : (78336, 2)
feature extraction length and channel : (325, 39)
pemrosesan.wav
signal length and channel : (72576, 2)
feature extraction length and channel : (301, 39)
```

import dictionary sebagai dataset

```
In [192...]
folder_path = './dictionary/'
test_template = []
if os.path.exists(folder_path) and os.path.isdir(folder_path):
    # List all files in the folder
    files = os.listdir(folder_path)

    # Iterate through the files
    for file_name in files:
        # You can process each file here, for example, print the file name
        print(file_name)
        (rate,sig) = wav.read(folder_path + file_name)
        print("signal length and channel : ", sig.shape )
        mfcc_feat = mfcc(sig,rate,nfilt = 39,numcep = 39,nfft = 2048)

        print("feature extraction length and channel : ", mfcc_feat.shape)
        test_template.append([file_name[:-4],mfcc_feat])
```

```
tugas.wav
signal length and channel : (61056, 2)
feature extraction length and channel : (253, 39)
merupakan.wav
signal length and channel : (70272, 2)
feature extraction length and channel : (292, 39)
kuliah.wav
signal length and channel : (81792, 2)
feature extraction length and channel : (340, 39)
kecil.wav
signal length and channel : (56448, 2)
feature extraction length and channel : (234, 39)
ini.wav
signal length and channel : (61056, 2)
feature extraction length and channel : (253, 39)
dictionary.wav
signal length and channel : (87552, 2)
feature extraction length and channel : (364, 39)
halo.wav
signal length and channel : (56448, 2)
feature extraction length and channel : (234, 39)
tambahan.wav
signal length and channel : (73728, 2)
feature extraction length and channel : (306, 39)
suara.wav
signal length and channel : (70272, 2)
feature extraction length and channel : (292, 39)
pemrosesan.wav
signal length and channel : (72576, 2)
feature extraction length and channel : (301, 39)
.ipynb_checkpoints
```

```
-----
IsADirectoryError                                     Traceback (most recent call last)
<ipython-input-192-d0b302a78bb2> in <cell line: 3>()
      9         # You can process each file here, for example, print the file name
     10        print(file_name)
--> 11        (rate,sig) = wav.read(folder_path + file_name)
     12        print("signal length and channel : ", sig.shape )
     13        mfcc_feat = mfcc(sig,rate,nfilt = 39,numcep = 39,nfft =2048)

/usr/local/lib/python3.10/dist-packages/scipy/io/wavfile.py in read(filename, mmap)
    645         mmap = False
    646     else:
--> 647         fid = open(filename, 'rb')
    648
    649     try:
```

IsADirectoryError: [Errno 21] Is a directory: './dictionary/.ipynb_checkpoints'

def prosedur untuk testing dan calc accuracy

In [193...]

```
def testSpeech(testset, template):
    truepos = 0
    for i in testset:
        bestClass = template[0][0]
        bestScore = 99999999.0
        for j in template:
            C = compute_cost_matrix(input_array=i[1], template=j[1])
            D = compute_accumulated_cost_matrix(C)
            # print(D[-1, -1])
            if (bestScore > D[-1, -1]).any():
```

```
        bestScore = D[-1, -1]
        bestClass = j[0]
        print("groundtruth : ", i[0])
        print("Predicted : ", bestClass)
        if(i[0] == bestClass):
            truepos += 1
        print("Score : ", bestScore)
        # print("DTW calculation using Library:", fastdtw(i[1].T, j[1].T, dist=euclidean))
        print("-----")
        acc = truepos/len(testset)
        print("accuracy : ", acc)
    return acc
```

Test pake suara nando

In [195...]

```
print(testSpeech(test_nando, template))
```

```
groundtruth : tugas
Predicted : tugas
Score : 17983.902183685215
-----
groundtruth : merupakan
Predicted : tambahan
Score : 17249.732581690678
-----
groundtruth : kuliah
Predicted : tambahan
Score : 18874.232399604327
-----
groundtruth : kecil
Predicted : tambahan
Score : 25274.01227611414
-----
groundtruth : ini
Predicted : tambahan
Score : 19406.35150214916
-----
groundtruth : dictionary
Predicted : dictionary
Score : 19840.879773246004
-----
groundtruth : halo
Predicted : tambahan
Score : 19443.043777487652
-----
groundtruth : tambahan
Predicted : tambahan
Score : 15322.35291049979
-----
groundtruth : suara
Predicted : tambahan
Score : 22717.07377154172
-----
groundtruth : pemrosesan
Predicted : tambahan
Score : 20912.951665499215
-----
accuracy : 0.3
0.3
```

Test pake suara rafli

In [196...]

```
print(testSpeech(test_rafli, template))

groundtruth : tugas
Predicted : halo
Score : 26707.485299908898
-----
groundtruth : merupakan
Predicted : halo
Score : 27221.776847280486
-----
groundtruth : kuliah
Predicted : halo
Score : 26456.12387813376
-----
groundtruth : kecil
Predicted : ini
Score : 26602.40984703177
-----
groundtruth : ini
Predicted : halo
Score : 28081.678683637438
-----
groundtruth : dictionary
Predicted : halo
Score : 27840.5405651443
-----
groundtruth : halo
Predicted : halo
Score : 28165.0635320707
-----
groundtruth : tambahan
Predicted : halo
Score : 27075.073191979835
-----
groundtruth : suara
Predicted : halo
Score : 25544.781633270613
-----
groundtruth : pemrosesan
Predicted : halo
Score : 26707.89421025205
-----
accuracy : 0.1
0.1
```

Test pake suara ronggur

In [197...]

```
print(testSpeech(test_ronggur, template))
```

```
groundtruth : tugas
Predicted : kecil
Score : 26741.52633176501
-----
groundtruth : merupakan
Predicted : merupakan
Score : 33083.98182293037
-----
groundtruth : kuliah
Predicted : kecil
Score : 28341.532111976652
-----
groundtruth : kecil
Predicted : kecil
Score : 30824.46344478791
-----
groundtruth : ini
Predicted : kecil
Score : 31647.313078436764
-----
groundtruth : dictionary
Predicted : kecil
Score : 34727.897348163715
-----
groundtruth : halo
Predicted : ini
Score : 42543.77701857394
-----
groundtruth : tambahan
Predicted : halo
Score : 35909.14760695019
-----
groundtruth : suara
Predicted : kecil
Score : 30473.62253096967
-----
groundtruth : pemrosesan
Predicted : kecil
Score : 32220.77128617396
-----
accuracy : 0.2
0.2
```

Test pake templatanya (harusnya acc 100% dan score 0)

```
In [199]: print(testSpeech(test_template, template))
```

```
groundtruth : tugas
Predicted : tugas
Score : 0.0
-----
groundtruth : merupakan
Predicted : merupakan
Score : 0.0
-----
groundtruth : kuliah
Predicted : kuliah
Score : 0.0
-----
groundtruth : kecil
Predicted : kecil
Score : 0.0
-----
groundtruth : ini
Predicted : ini
Score : 0.0
-----
groundtruth : dictionary
Predicted : dictionary
Score : 0.0
-----
groundtruth : halo
Predicted : halo
Score : 0.0
-----
groundtruth : tambahan
Predicted : tambahan
Score : 0.0
-----
groundtruth : suara
Predicted : suara
Score : 0.0
-----
groundtruth : pemrosesan
Predicted : pemrosesan
Score : 0.0
-----
accuracy : 1.0
1.0
```

Template dibuat oleh nando pada hari yang berbeda

Akurasi

dataset nando : 0.3

dataset rafli : 0.1

dataset ronggur : 0.2