

LAPORAN TUGAS KECIL 2

IF2211 STRATEGI ALGORITMA

**“Implementasi Convex Hull untuk Visualisasi Tes Linear Separability
Dataset dengan Algoritma Divide and Conquer”**



NIM : 13520132
Nama : Januar Budi Ghifari
Kelas : K03

PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TEKNIK ELEKTRO DAN
INFORMATIKA INSTITUT TEKNOLOGI BANDUNG 2021

A. Algoritma Divide and Conquer

Program ini dibuat untuk menyelesaikan Visualisasi Tes Linear Separability Dataset dengan Algoritma Divide and Conquer menggunakan Bahasa pemrograman Python. Langkah yang saya gunakan untuk menyelesaikan visualisasi ini adalah dengan membandingkan jarak semua titik-titik dengan garis pembagi untuk mencari titik pembentuk convex yang mungkin. Berikut deskripsi singkat cara kerja program saya.

1. Program berjalan dengan memanggil fungsi `ConvexHull()` yang akan memanggil fungsi `divideAndConquer()` secara rekursif dengan menerapkan algoritma Divide and Conquer
2. Sebagai inisiasi, data set akan dibelah menjadi 2 bagian secara horizontal dengan mencari 2 titik terjauh. Setelah itu `ConvexHull()` akan memanggil `divideAndConquer` 2 kali untuk menyelesaikan bagian atas dan bawah.
3. fungsi `divideAndConquer` akan membandingkan jarak semua titik yang ada dengan garis pembagi lalu mengambil titik terjauh sebagai titik yang akan membentuk 2 garis baru dari ujung-ujung garis pembagi. Nantinya 2 garis itu akan diproses lagi dengan fungsi `divideAndConquer()`.
4. fungsi berjalan terus secara rekursif sampai tidak ada lagi titik terjauh yang tersedia sehingga pada akhirnya terkumpul garis-garis pembentuk convex.

B. Source Code Program

main

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
print("1. iris dataset")
print("2. breast cancer dataset")
print("3. wine dataset")
print("Pilih dataset mana yang akan digunakan: ")
indata = int(input())
while (indata != 1) and (indata != 2) and (indata != 3):
    indata = (input(int("Pilih dataset mana yang akan digunakan: ")))

if indata == 1:
    data = datasets.load_iris()
elif indata == 2:
    data = datasets.load_breast_cancer()
elif indata == 3:
    data = datasets.load_wine()

df = pd.DataFrame(data.data, columns=data.feature_names)
df['Target'] = pd.DataFrame(data.target)
print(df.shape)
df.head()
```

```
import myConvexHull
plt.figure(figsize = (10, 6))
colors = ['b', 'r', 'g']
plt.title('Petal Width vs Petal Length')
plt.xlabel(data.feature_names[0])
plt.ylabel(data.feature_names[1])
for i in range(len(data.target_names)):
    bucket = df[df['Target'] == i]
    bucket = bucket.iloc[:, [0, 1]].values
    hull = myConvexHull.ConvexHull(bucket) #bagian ini diganti dengan hasil implementasi
    plt.scatter(bucket[:, 0], bucket[:, 1], label=data.target_names[i])
    for simplex in hull:
        plt.plot(bucket[simplex, 0], bucket[simplex, 1], colors[i])
plt.legend()
```

myConvexHull

```
import numpy as np

def jarakTitikKeGaris(p1, p2, p3): #fungsi untuk mencari panjang jarak sebuah titik ke garis
    #menerima input 3 buah titik dengan p1 dan p2 merupakan ujung-ujung garis, dan p3 adalah titik
    p1 = np.asarray(p1)
    p2 = np.asarray(p2)
    p3 = np.asarray(p3)

    jarak = np.cross(p2-p1, p3-p1)/np.linalg.norm(p2-p1)

    return jarak

def divideAndConquer(bucket, kiri, kanan, batas, ans):
    #fungsi menentukan titik yang akan menjadi convexhull, fungsi divideAndConquer ini berjalan secara rekursif
    dotKiri = bucket[kiri]
    dotKanan = bucket[kanan]

    mid = -1 #akan menjadi titik terjauh dari 2 garis
    jarakTerjauh = 0

    p1 = (dotKiri[0], dotKiri[1])
    p2 = (dotKanan[0], dotKanan[1])

    for i in range(len(bucket)) :
        p3 = (bucket[i][0], bucket[i][1])
        jarak = jarakTitikKeGaris(p1, p2, p3)

        if jarak * batas >= 0 and abs(jarak) > jarakTerjauh :
            #batas digunakan agar pencarian titik terjauh selalu mengarah "keluar" garis
            mid = i
            jarakTerjauh = abs(jarak)

    if mid == -1 : #apabila tidak terdapat titik lagi yang dapat dicari
        newLine = [kiri, kanan]
        if newLine not in ans :
            ans.append(newLine)
        return ans
    #rekursif menggunakan titik terjauh dari 2 garis yang kemudian memotong garis menjadi 3 titik (kiri, tengah, kanan)
    ans = divideAndConquer(bucket, kiri, mid, batas, ans)
    ans = divideAndConquer(bucket, mid, kanan, batas, ans)

    return ans

def ConvexHull(bucket) :
    idxXkiri = 0
    idxXkanan = 0
    ans = []

    for i in range(len(bucket)) :
        #sebagai inisiasi untuk memotong data menjadi 2 bagian dengan mencari 2 titik terjauh secara horizontal (atas dan bawah)
        if bucket[i][0] < bucket[idxXkiri][0] :
            idxXkiri = i
        if bucket[i][0] > bucket[idxXkanan][0] :
            idxXkanan = i

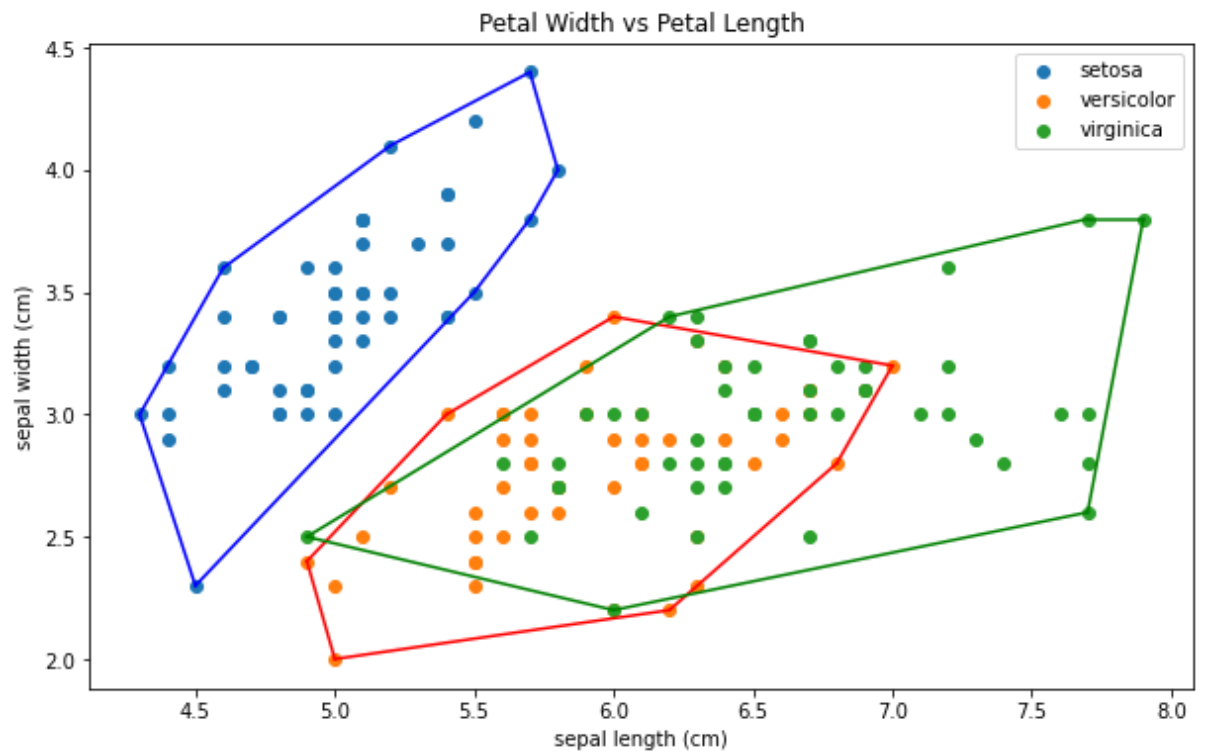
    # -1 dan 1 akan menjadi batasan pencarian agar pencarian selalu mengarah "keluar" garis
    ans = divideAndConquer(bucket, idxXkiri, idxXkanan, 1, ans)
    # 1 untuk mencari di bagian atas data yang sudah dibagi 2
    ans = divideAndConquer(bucket, idxXkiri, idxXkanan, -1, ans)
    # -1 untuk mencari di bagian bawah data yang sudah dibagi 2

    return ans
```

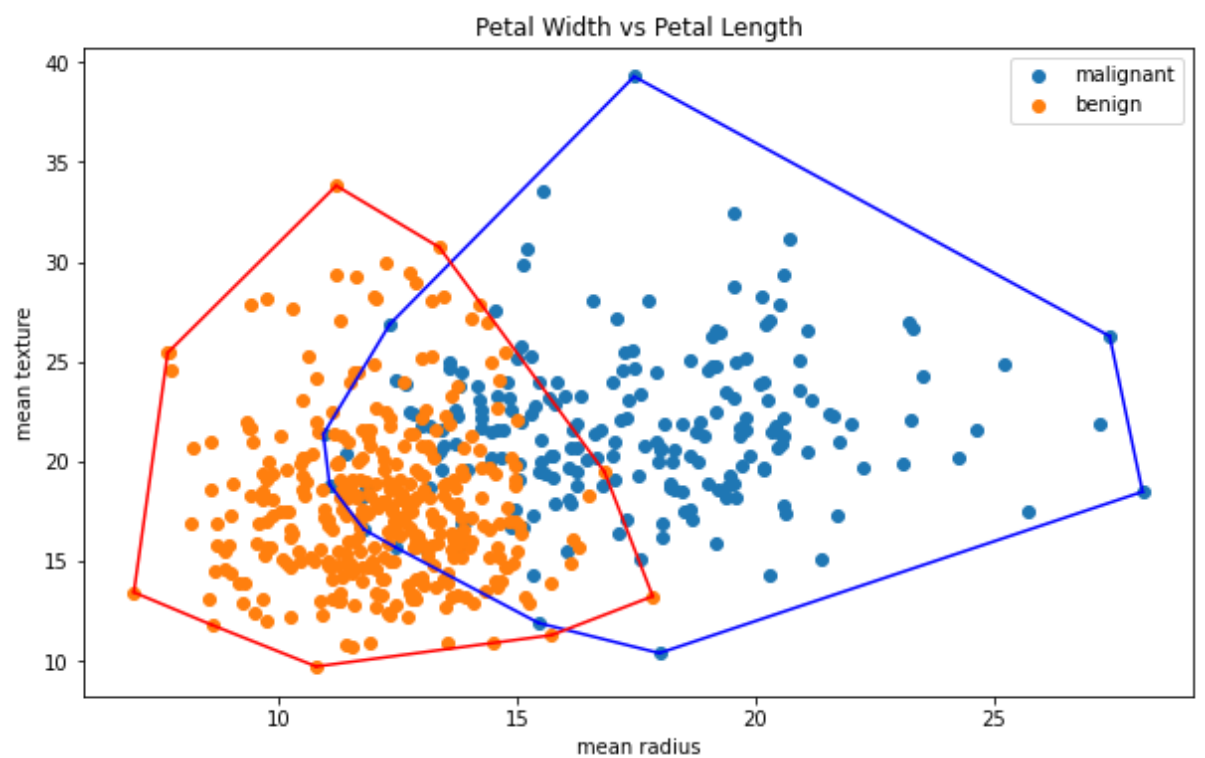
C. Output

Output di bawah didapat dari 3 dataset berbeda dari situs https://scikit-learn.org/stable/datasets/toy_dataset.html

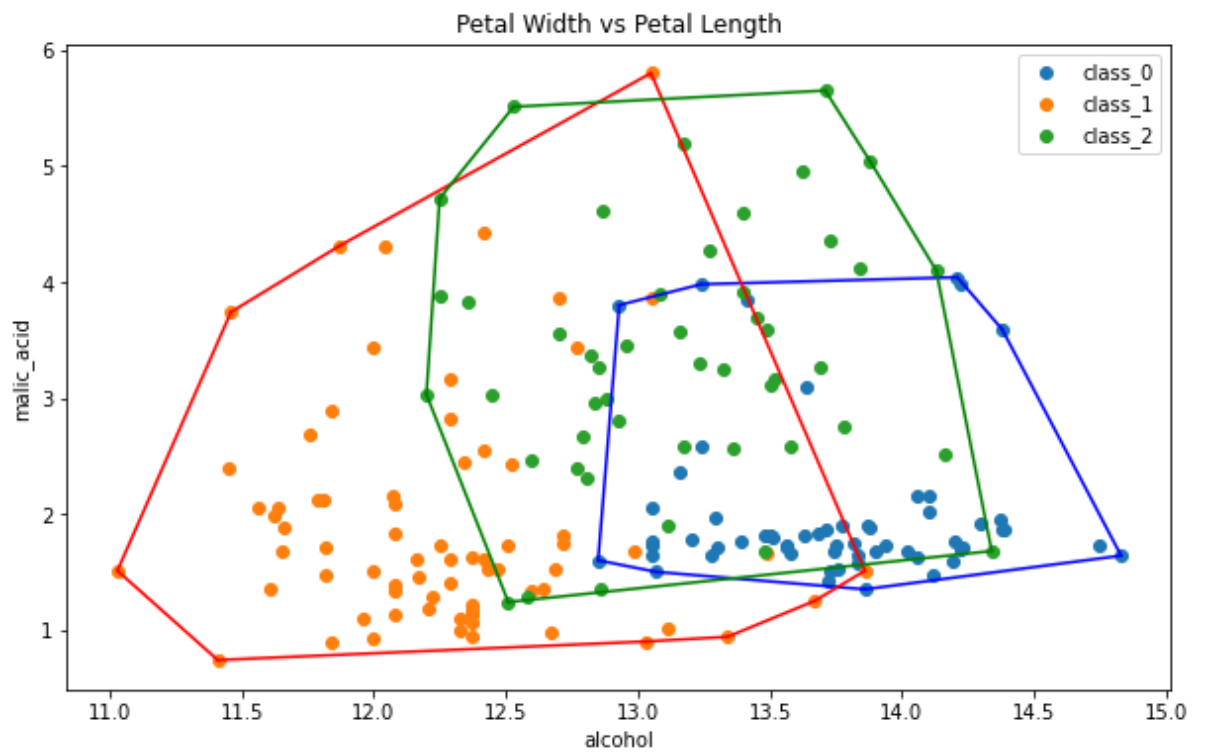
dataset iris



dataset breast_cancer



dataset wine



Poin	Ya	Tidak
1. Pustaka myConvexHull berhasil dibuat dan tidak ada kesalahan	√	
2. Convex hull yang dihasilkan sudah benar	√	
3. Pustaka myConvexHull dapat digunakan untuk menampilkan convex hull setiap label dengan warna yang berbeda.	√	
4. Bonus: program dapat menerima input dan menuliskan output untuk dataset lainnya.	√	