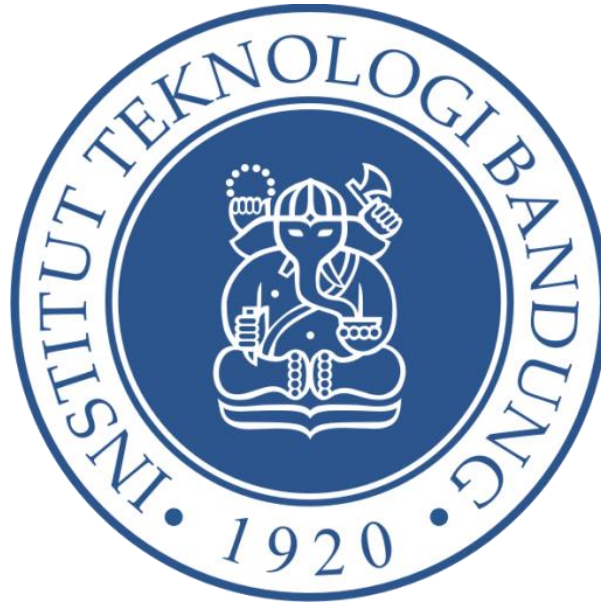


TUGAS KECIL 2 IF2211 Strategi Algoritma

Semester 2 Tahun 2021/2022

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



**Disusun oleh**

M Syahrul Surya Putra - 13520161

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

**2022**

#### A. Algoritma *Divide and Conquer*

Algoritma yang digunakan pada tugas kecil pencarian *convex hull* ini ialah *divide and conquer*. Adapaun langkah-langkah adalah sebagai berikut:

1. Pertama-tama, program akan meminta user untuk memilih dataset apa yang ingin dijalankan. Pada program ini sendiri, hanya disediakan 4 dataset, yaitu Iris Sepal, Iris Petal, Wine dan Breast Cancer
2. Setelah memilih dataset, program akan membaca datanya terlebih dahulu, kemudian akan di ubah ke list untuk memudahkan pemrosesan data dan peplottan grafiknya nanti
3. Setelah kedua langkah di atas selesai, hal pertama yang akan dilakukan oleh program adalah mencari 2 titik ekstrim dari data yang ada menggunakan fungsi *getExtreme*. Fungsi ini akan memberikan data berupa tuple(min, max). Dimana min ini merupakan titik dengan absis paling kecil dan sebaliknya, max merupakan titik dengan absis paling besar
4. Setelah didapatkan titik ekstrim yang pertama, dicarilah kandidat untuk *convex hull* dimulai dari bagian kiri/atas dari titik ekstrim yang telah didapatkan. Setelah didapatkan kandidat dari bagian kiri/atas, dilanjutkan dengan pencarian kandidat *convex hull* pada bagian kanan/bawah dari titik ekstrim. Langkah ini dilakukan dengan rekursif
5. Untuk pencarian kandidat *convex hull*nya sendiri, program akan mencari indeks dari titik yang terjauh dari garis lurus terakhir.
6. Ketika ditemukan satu titik terjauh dari garis lurus terakhir, program akan memanggil fungsi *findHull* lagi untuk mencari apakah masih ada titik dari garis yang baru saja ditemukan. Proses ini sama seperti langkah 4, program akan memulai dari sisi kiri/atas, ketika telah selesai, dilanjutkan pada bagian kanan/bawah
7. Langkah 5 dan 6 ini akan terus berulang, sampai tidak ditemukan titik terjauh dari garis terakhir
8. Setelah didapatkan list *convex hull*, program akan mengembalikan list tersebut untuk di plot

## B. Source Code

- visualizer.py

```
#visualisasi hasil ConvexHull
import matplotlib.pyplot as plt
from myConvexHull import ConvexHull

import numpy as np
import pandas as pd
from sklearn import datasets

def menu():
    print("===== Pilih Data =====")
    print("1. Iris Sepal")
    print("2. Iris Petal")
    print("3. Wine")
    print("4. Breast Cancer")

menu()
opt = int(input("Pilihan anda >> "))
colors = ['b','g','r', 'c', 'm', 'y', 'k']

if opt == 1:
    data = datasets.load_iris()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    print(df.shape)
    print("Feature: ", data.feature_names)
    print("Target: ", data.target_names)
    df.head()

    plt.figure(figsize = (10, 6))
    plt.title('Sepal Width vs Sepal 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 = ConvexHull(bucket) #bagian ini diganti dengan hasil implementasi
        # ConvexHull Divide & Conquer

        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 % len(colors)])

plt.legend()
plt.savefig("./out/iris_sepal.png")
plt.show()

elif opt == 2:
    data = datasets.load_iris()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    print(df.shape)
    print("Feature: ", data.feature_names)
    print("Target: ", data.target_names)
    df.head()

    plt.figure(figsize = (10, 6))
    plt.title('Petal Width vs Petal Length')
    plt.xlabel(data.feature_names[2])
    plt.ylabel(data.feature_names[3])
    for i in range(len(data.target_names)):
        bucket = df[df['Target'] == i]
        bucket = bucket.iloc[:,[2,3]].values

        hull = ConvexHull(bucket) #bagian ini diganti dengan hasil
implementasi
        # ConvexHull Divide & Conquer

        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 % len(colors)])

plt.legend()
plt.savefig("./out/iris_petal.png")
plt.show()

elif opt == 3:
    data = datasets.load_wine()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    print(df.shape)
    print("Feature: ", data.feature_names)

```

```

print("Target: ", data.target_names)
df.head()

plt.figure(figsize = (10, 6))
plt.title('Alcohol vs Malic Acid')
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 = ConvexHull(bucket) #bagian ini diganti dengan hasil implementasi
    # ConvexHull Divide & Conquer

    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 % len(colors)])

plt.legend()
plt.savefig("./out/wine.png")
plt.show()

elif opt == 4:
    data = datasets.load_breast_cancer()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    print(df.shape)
    print("Feature: ", data.feature_names)
    print("Target: ", data.target_names)
    df.head()

    plt.figure(figsize = (10, 6))
    plt.title('Mean Smoothness vs Mean Compactness')
    plt.xlabel(data.feature_names[4])
    plt.ylabel(data.feature_names[5])
    for i in range(len(data.target_names)):
        bucket = df[df['Target'] == i]
        bucket = bucket.iloc[:,[4,5]].values

        hull = ConvexHull(bucket) #bagian ini diganti dengan hasil implementasi
        # ConvexHull Divide & Conquer

```

```

        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 % len(colors)])

plt.legend()
plt.savefig("./out/breast_cancer.png")
plt.show()
else:
    print("\nInput invalid.")

```

- myConvexHull.py

```

def getExtreme(points):
    # [x, y]
    min, max = points[0], points[0]
    minIdx, maxIdx = 0, 0

    for i in range(1, len(points)):
        if (points[i][0] < min[0]):
            min = points[i]
            minIdx = i
        if (points[i][0] > max[0]):
            max = points[i]
            maxIdx = i

    return minIdx, maxIdx

# Fungsi untuk mencari jarak dari garis yang dibentuk oleh p1 dan
p2
# dengan titik p3
def distance(p1, p2, p3):
    a = (p2[0] - p1[0]) * (p1[1] - p3[1])
    b = (p1[0] - p3[0]) * (p2[1] - p1[1])
    d = ((p2[0]-p1[0])**2 + (p2[1] - p1[1])**2)**(1/2)
    return((a - b) / d)

# Mencari titik terjauh dari garis p1p2
# Urutan p1 dan p2 berpengaruh
def getFarthest(p1, p2, points):
    max = 0
    maxIdx = -1

```

```

    for i in range(len(points)):
        dist = distance(p1, p2, points[i])
        if (max < dist):
            max = dist
            maxIdx = i

    return maxIdx

# Rekursi pencarian convexhull
def findHull(p1Idx, p2Idx, points, hull):
    maxIdx = getFarthest(points[p1Idx], points[p2Idx], points)

    # Tidak ditemukan titik di luar titik p1 dan p2
    if (maxIdx == -1):
        hull.append([p1Idx, p2Idx])
        return hull

    hull = findHull(p1Idx, maxIdx, points, hull)
    hull = findHull(maxIdx, p2Idx, points, hull)
    return hull

def ConvexHull(points):
    hull = [] # Hanya menerima index untuk memudahkan plotting

    points = points.tolist()
    p1Idx, pnIdx = getExtreme(points)

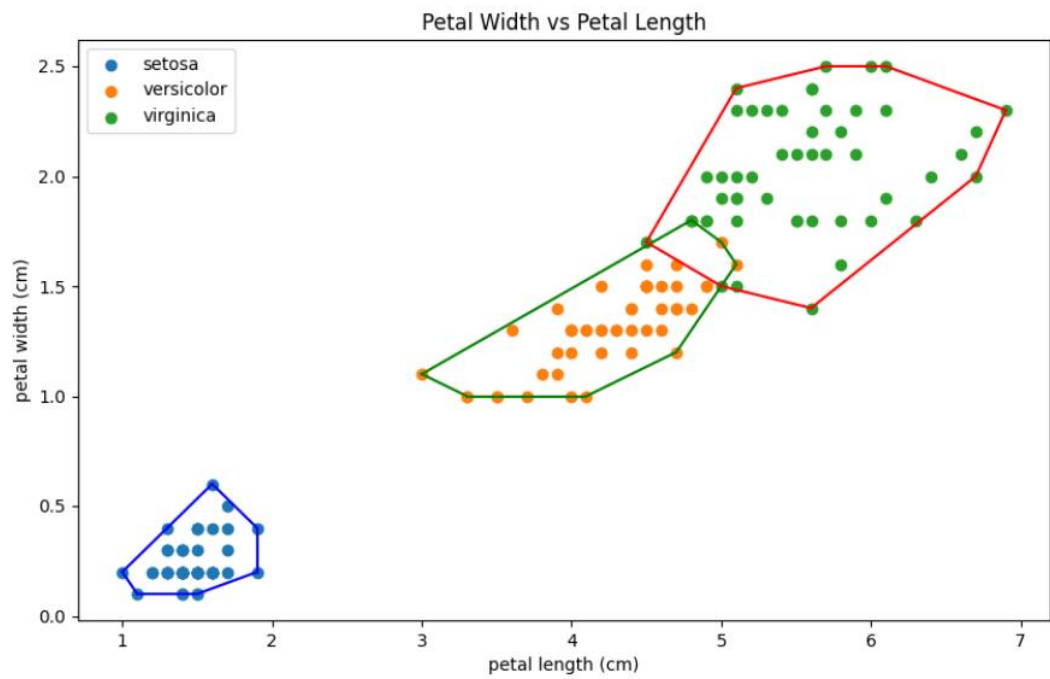
    hull = findHull(p1Idx, pnIdx, points, hull)
    hull = findHull(pnIdx, p1Idx, points, hull)

    return hull

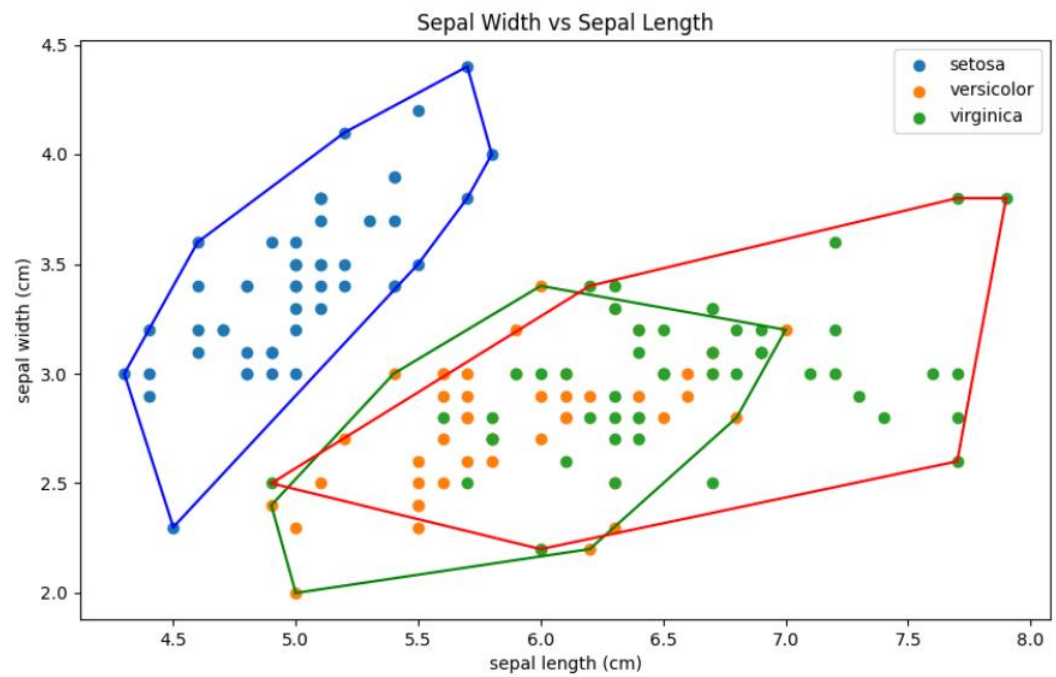
```

### C. Screenshot Output

- Iris Petal

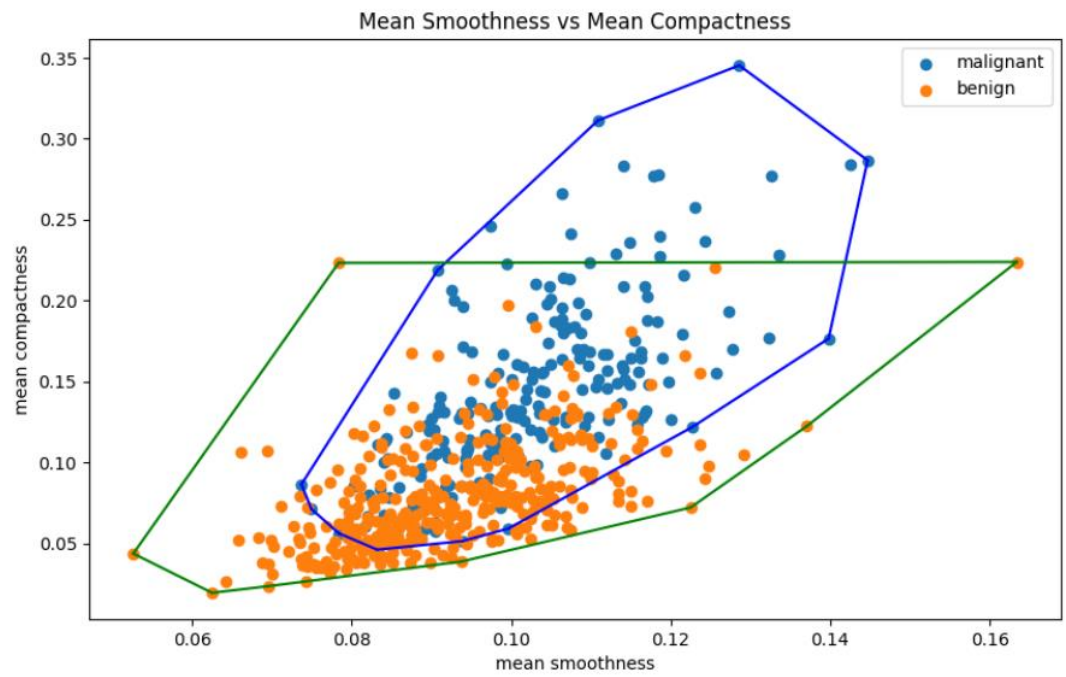


- Iris Sepal

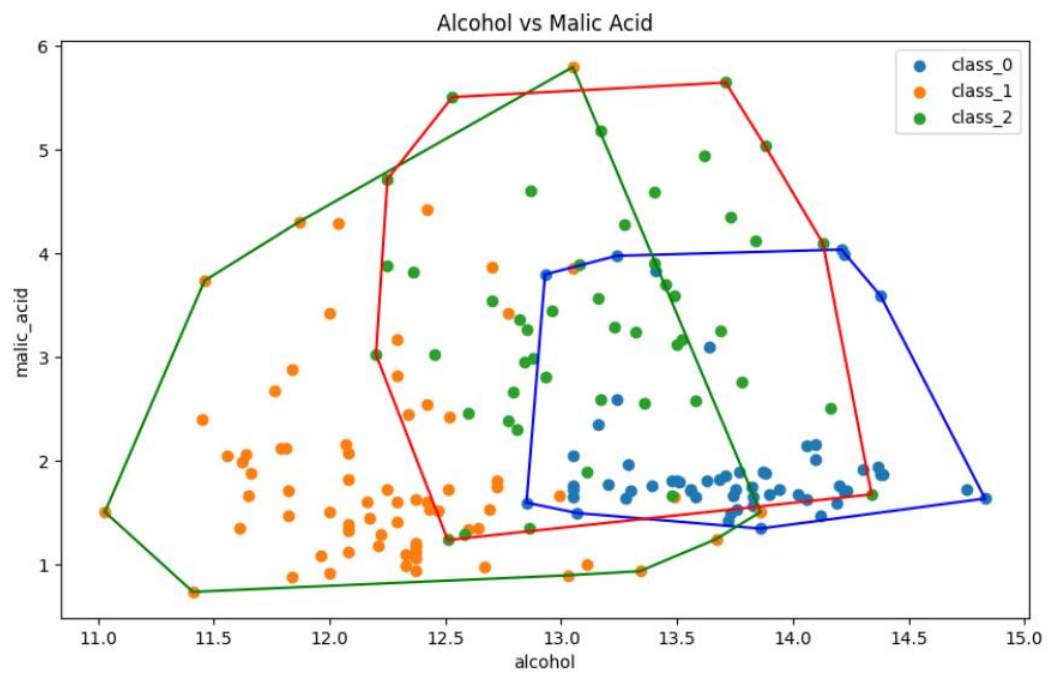




- Breast\_Cancer



- Wine



#### D. Lain-lain

Link Repository: <https://github.com/msyahrulsp/convex-hull-boogaloo>

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