

Laporan Tugas Kecil 2
IF 2211 Strategi Algoritma
Implementasi Convex Hull untuk Visualisasi Tes
Linear Separability Dataset* dengan Algoritma *Divide
and Conquer



Disusun Oleh :
Daffa Romyz Aufa 13520162

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

A. Algoritma *Divide and Conquer*

berikut implementasi algoritma *divide and conquer* untuk penyelesaian untuk Visualisasi *Tes Linear Separability Dataset*.

1. Menentukan titik minimum A dan titik maksimum B berdasarkan nilai absis. Jika nilai absis sama maka yang diambil adalah yang memiliki nilai ordinat yang lebih besar. Titik A dan B akan dimasukkan sebagai titik-titik pembangun *convex hull*
2. Bagi data titik menjadi 2 grup berdasarkan garis AB. Titik yang mengenai garis AB diabaikan. Misalkan, G1 adalah grup yang berada di kanan garis AB dan G2 adalah grup yang berada di kanan garis BA.
3. Cari titik-titik pembangun *convex hull* untuk G1 berdasarkan garis AB dan titik-titik pembangun *convex hull* untuk G1 berdasarkan garis BA.
4. Untuk sebuah grup (misalkan G1) akan dicari titik-titik pembangun *convex hull* berdasarkan garis AB, terdapat 2 kemungkinan:
 - Jika G1 kosong, maka titik-titik pembangun *convex hull* tidak berubah
 - Jika G1 tidak kosong maka akan dicari titik C terjauh dari garis AB. Titik C akan dimasukkan sebagai titik-titik pembangun *convex hull*. Abaikan titik-titik yang ada dalam segitiga ABC. Bagi G1 menjadi 2 grup. G11 adalah grup yang berada di kanan garis AC. G12 adalah grup yang berada di kanan garis CB. Lalu lakukan langkah 3 dan 4 pada grup baru ini.

B. Source Code Program

1. `input_data.py`

```
import pandas as pd
from sklearn import datasets

def data(select):
    if select == 1:
        print("Data = iris")
        data = datasets.load_iris()
    elif select == 2:
        print("Data = wine")
        data = datasets.load_wine()
    elif select == 3:
        print("Data = breast_cancer")
        data = datasets.load_breast_cancer()
    else:
        print("Data = iris")
        data = datasets.load_iris()
    return data
```

```
#create a DataFrame
def dataframe(data):
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    df.head()
    return df
```

2. dnc.py

```
import numpy as np
import math
```

```
def convexhull(bucket):
    arr = np.array([[0, 0]])
    minp = np.array(bucket[0])
    maxp = np.array(bucket[0])

    # mencari titik ekstrim
    for i in range(len(bucket)):
        if minp[0] >= bucket[i][0]:
            if (minp[0] == bucket[i][0]) and (minp[1] < bucket[i][1]):
                minp = bucket[i]
            else:
                minp = bucket[i]
        if maxp[0] <= bucket[i][0]:
            if (maxp[0] == bucket[i][0]) and (maxp[1] < bucket[i][1]):
                maxp = bucket[i]
            else:
                maxp = bucket[i]

    # membagi titik-titik menjadi 2 grup
    s1 = np.array([[0, 0]])
    s2 = np.array([[0, 0]])

    for i in bucket:
        if det(minp[0], minp[1], maxp[0], maxp[1], i[0], i[1]) > 0 and \
            not(minp[0] == i[0] and minp[1] == i[1]) and not(maxp[0] == i[0] and maxp[1] == i[1]):
            s1 = np.append(s1, [i], axis=0)
        if det(maxp[0], maxp[1], minp[0], minp[1], i[0], i[1]) > 0 and \
            not(minp[0] == i[0] and minp[1] == i[1]) and not(maxp[0] == i[0] and maxp[1] == i[1]):
            s2 = np.append(s2, [i], axis=0)
    s1 = np.delete(s1, 0, 0)
    s2 = np.delete(s2, 0, 0)

    # rekurens
```

```

arr = np.append(arr, [minp], axis=0)
arr = np.delete(arr, 0, 0)
arr = findpoints(arr, s1, minp[0], minp[1], maxp[0], maxp[1])
arr = np.append(arr, [maxp], axis=0)
arr = findpoints(arr, s2, maxp[0], maxp[1], minp[0], minp[1])
arr = np.append(arr, [minp], axis=0)
return arr

```

```

def det(x1, y1, x2, y2, x3, y3):
    # menentukan posisi titik
    # det = 0 jika berada dalam garis, det > 0 jika di kanan garis, det < 0 jika dikiri garis
    return x1*y2 + x3*y1 + x2*y3 - x3*y2 - x2*y1 - x1*y3

```

```

def rangepl(x1, y1, x2, y2, x0, y0):
    # menentukan jarak titik (x0,y0) ke garis (x1,y1)(x2,y2)
    return abs(((x2 - x1)*(y1-y0) - (x1 - x0)*(y2-y1))/(math.sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1))))

```

```

def findpoints(arr, bucket, x1, y1, x2, y2):
    s1 = np.array([[0, 0]])
    s2 = np.array([[0, 0]])

    if bucket.size == 0: #basis : jika bucket kosong maka arr tidak berubah
        return arr
    else:
        # mencari titik paling jauh dari garis (x1,y1)(x2,y2)
        C = bucket[0]
        for i in bucket:
            if rangepl(x1, y1, x2, y2, i[0], i[1]) > rangepl(x1, y1, x2, y2, C[0], C[1]):
                C = i

        # membagi titik-titik menjadi 2 grup
        for i in bucket:
            if det(x1, y1, C[0], C[1], i[0], i[1]) > 0 and \
                not(x1 == i[0] and y1 == i[1]) and not(C[0] == i[0] and C[1] == i[1]):
                s1 = np.append(s1, [i], axis=0)
            if det(C[0], C[1], x2, y2, i[0], i[1]) > 0 and \
                not(x2 == i[0] and y2 == i[1]) and not(C[0] == i[0] and C[1] == i[1]):
                s2 = np.append(s2, [i], axis=0)
        s1 = np.delete(s1, 0, 0)
        s2 = np.delete(s2, 0, 0)

```

```

# rekurens
arr = findpoints(arr, s1, x1, y1, C[0], C[1])
arr = np.append(arr, [C], axis=0)
arr = findpoints(arr, s2, C[0], C[1], x2, y2)
return arr

```

3. main.py

```

# visualisasi hasil ConvexHull
import matplotlib.pyplot as plt
import input_data
import dnc

print("Select Data : ")
print("default -> iris")
print("1 -> iris")
print("2 -> wine")
print("3 -> breast_cancer")
select = int(input(">>>"))

data = input_data.data(select)
df = input_data.dataframe(data)

plt.figure(figsize=(10, 6))
colors = ['b', 'r', 'g', 'c', 'm', 'y', 'k', 'w']

if select == 1:
    plt.title('iris')
elif select == 2:
    plt.title('wine')
elif select == 3:
    plt.title('breast_cancer')
else:
    plt.title('iris')

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 = dnc.convexhull(bucket)
    plt.scatter(bucket[:, 0], bucket[:, 1], label=data.target_names[i])
    plt.plot(hull[:, 0], hull[:, 1], colors[i])
plt.legend()

```

```

print("Save figure to bin folder? y/[n]")
save = input(">>>")
if save == 'y':
    plt.savefig("../bin/figure.png")
    print("Figure saved as \"Figure.png\" in bin folder")

```

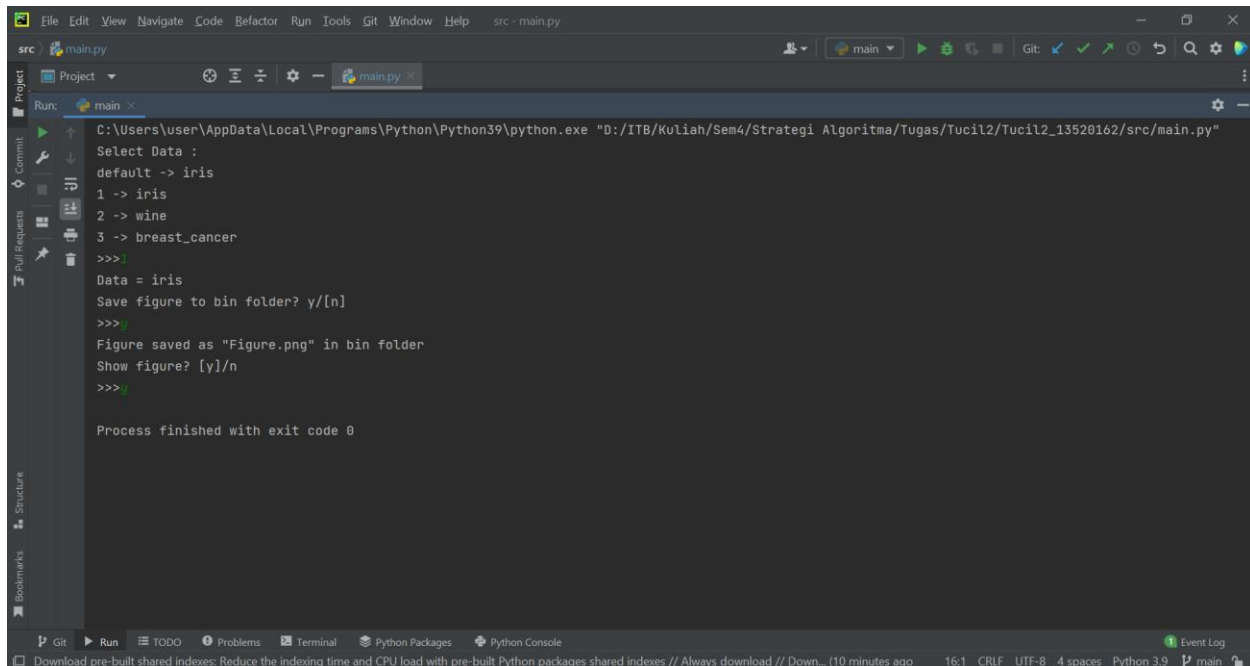
```

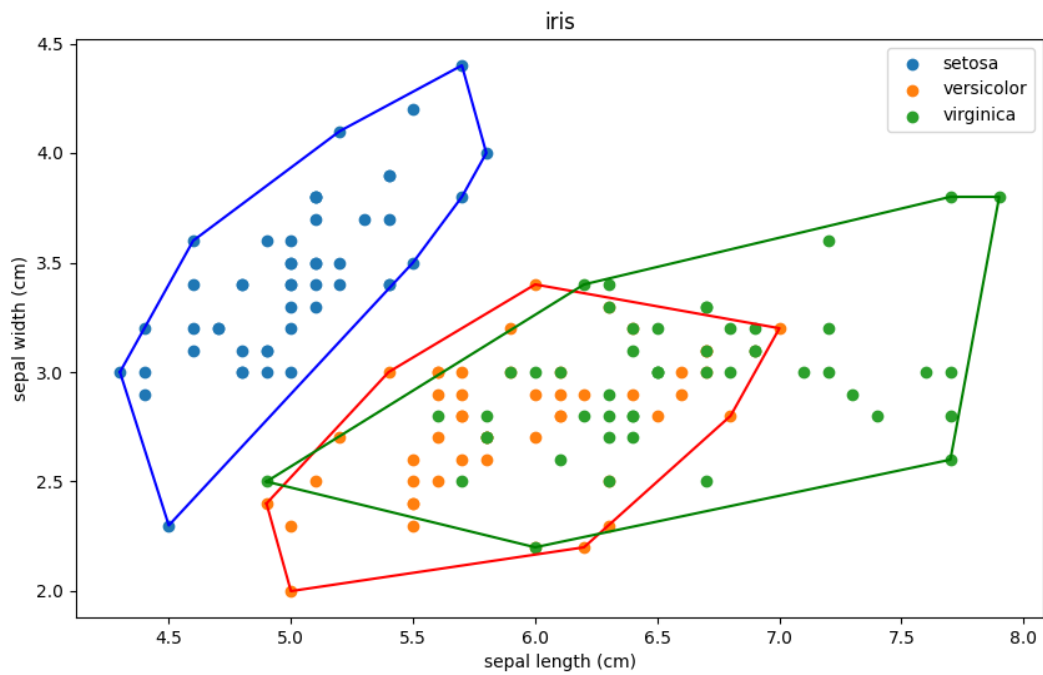
print("Show figure? [y]/n")
show = input(">>>")
if show == 'n':
    exit()
else:
    plt.show()

```

C. Screenshoot testing

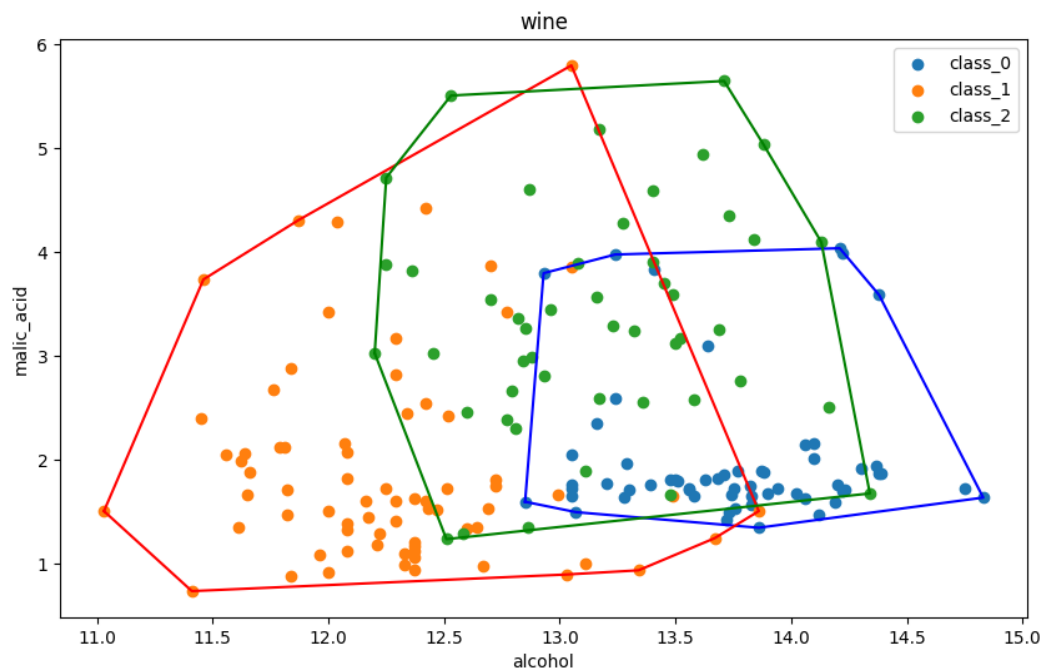
1. Data iris





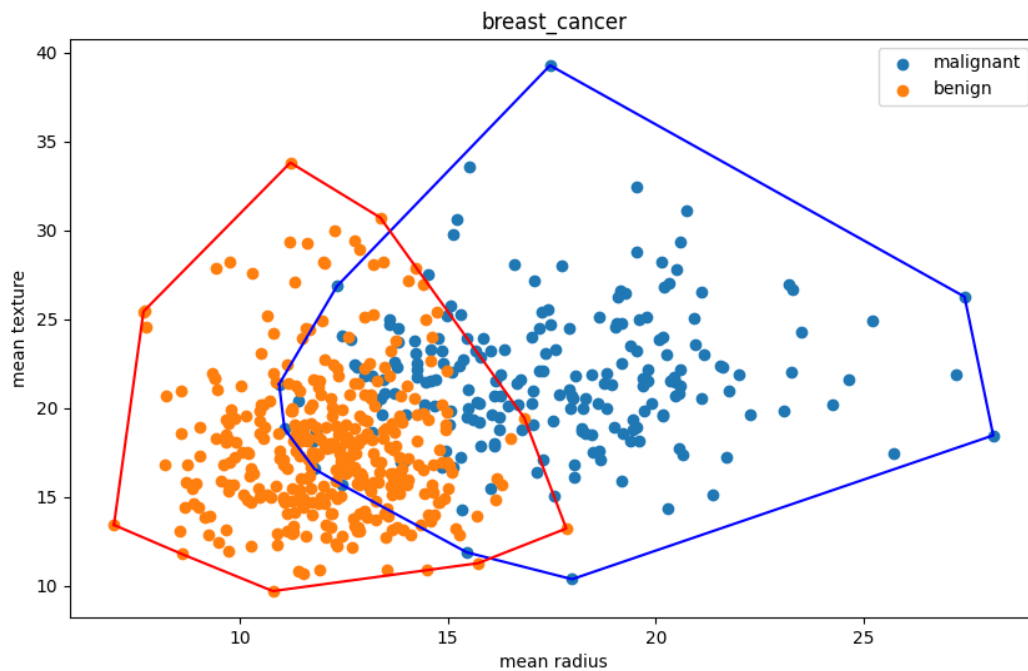
2. Data wine

```
File Edit View Navigate Code Refactor Run Tools Git Window Help src - main.py
src main.py
Project main.py
Run: C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "D:/ITB/Kuliah/Sem4/Strategi Algoritma/Tugas/Tucil2/Tucil2_13520162/src/main.py"
Select Data :
default -> iris
1 -> iris
2 -> wine
3 -> breast_cancer
>>>2
Data = wine
Save figure to bin folder? y/[n]
>>>y
Figure saved as "Figure.png" in bin folder
Show figure? [y]/n
>>>y
Process finished with exit code 0
Git Run TODO Problems Terminal Python Packages Python Console
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Down... (11 minutes ago) 16:1 CRLF UTF-8 4 spaces Python 3.9 main
```



3. Data breast_cancer

```
File Edit View Navigate Code Refactor Run Tools Git Window Help src - main.py
src main.py
Project
Run: C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "D:/ITB/Kuliah/Sem4/Strategi Algoritma/Tugas/Tucil2/Tucil2_13520162/src/main.py"
Select Data :
default -> iris
1 -> iris
2 -> wine
3 -> breast_cancer
>>>
Data = breast_cancer
Save figure to bin folder? y/[n]
>>>
Figure saved as "Figure.png" in bin folder
Show figure? [y]/n
>>>
Process finished with exit code 0
|
Git Run TODO Problems Terminal Python Packages Python Console
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Down... (12 minutes ago) 16:1 CRLF UTF-8 4 spaces Python 3.9 main
```

D. Link github berisi kode program

https://github.com/DaffaRomyz/Tucil2_13520162

Poin	Ya	Tidak
1. Pustaka <i>myConvexHull</i> berhasil dibuat dan tidak ada kesalahan	Ya	
2. <i>Convex hull</i> yang dihasilkan sudah benar	Ya	
3. Pustaka <i>myConvexHull</i> dapat digunakan untuk menampilkan <i>convex hull</i> setiap label dengan warna yang berbeda.	Ya	
	Ya	

4. Bonus: program dapat menerima input dan menuliskan output untuk dataset lainnya.		
---	--	--