

# **Tugas Kecil 1 IF2211 Strategi Algoritma:**

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



**Dibuat oleh:**

Rahmat Rafid Akbar

13520090

PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2022

## A. Algoritma Penentuan Convex Hull dari Himpunan Titik S

1. Mula-mula lakukan pengurutan pada data secara terpisah, pengurutan dilakukan terhadap sumbu x dan dilanjutkan terhadap sumbu y secara menaik. Data dapat disimpan di array N-dimensi. (Bukan termasuk algoritma *D&C*)
2. Tentukan 2 buah titik ekstrim pembentuk Garis-Pembagi, yaitu data elemen pertama ( $p_1$ ) dan data elemen terakhir ( $p_n$ ) dari data yang telah terurut.
3. Kumpulan titik selain titik ekstrim akan dipisahkan terhadap Garis-Pembagi menjadi 2 bagian himpunan. Ditentukan dengan mencari determinan dari 2 buah titik ekstrim ( $p_1$  dan  $p_2$ ) serta sebuah titik uji ( $p_3$ ), dengan rumus:

$$\det = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} = ((x_1 y_2) - (x_2 y_1)) + ((x_2 y_3) - (x_3 y_2)) - ((x_1 y_3) - (x_3 y_1))$$

Akan ada 3 nilai determinan:

- a)  $\det > 0$ , maka titik uji ( $p_3$ ) berada di sebelah kiri garis  $p_1 p_2$   $\rightarrow$  Himpunan  $S_1$
  - b)  $\det < 0$ , maka titik uji ( $p_3$ ) berada di sebelah kanan garis  $p_1 p_2$   $\rightarrow$  Himpunan  $S_2$
  - c)  $\det = 0$ , maka titik uji ( $p_3$ ) berada pada garis  $p_1 p_2$   $\rightarrow$  Abaikan
4. Untuk **SETIAP** bagian Himpunan  $S_n$  akan terdapat 2 buah kemungkinan:
    - a) Himpunan tersebut kosong, maka Garis-Pembagi himpunan tersebut adalah convex pada bagian himpunan tersebut. Lanjut ke langkah 5.
    - b) Himpunan tersebut tidak kosong, lanjut ke langkah 4a. s.d. langkah 4c.
  - 4a. Pilih sebuah titik ( $p_3$ ) pada himpunan  $S_n$  yang memiliki jarak ( $d$ ) terjauh dari Garis-Pembagi sebelumnya ( $p_1 p_2$ ). Ditentukan dengan rumus:

$$\vec{a} = p_2 - p_1 \quad ; \quad \vec{b} = p_1 - p_3 \quad ; \quad \text{norm}(\vec{p}) = \sqrt{x^2 + y^2}$$
$$d = \left| \frac{\text{norm}(\vec{a} \times \vec{b})}{\text{norm}(\vec{a})} \right|$$

Apabila terdapat beberapa titik dengan jarak terjauh ( $d$ ) yang sama, maka ditentukan dengan sudut ( $\theta$ ) terbesar yang ditentukan dengan rumus:

$$\vec{bc} = p_2 - p_1 \quad ; \quad \vec{ba} = p_3 - p_1 \quad ; \quad \text{norm}(\vec{p}) = \sqrt{x^2 + y^2}$$
$$\theta = \cos^{-1} \left( \frac{\vec{ba} \cdot \vec{bc}}{\text{norm}(\vec{ba}) * \text{norm}(\vec{bc})} \right)$$

- 4b. Bentuk 2 buah Garis-Pembagi baru (*LineL* & *LineR*). *LineL* dibentuk dari titik awal Garis-Pembagi sebelumnya ( $p_1$ ) dan titik terjauh ( $p_{max}$ ) atau  $p_1 p_{max}$ . *LineR* dibentuk dari titik terjauh ( $p_{max}$ ) dan titik akhir Garis-Pembagi sebelumnya ( $p_2$ ) atau  $p_{max} p_2$ .

**NOTE** : Perhatikan arah/ urutan pembuatan garis + akan terbentuk segitiga dari 2 Garis-Pembagi baru dan Garis Pembagi sebelumnya.

- 4c. Semua titik pada bagian himpunan ( $S_n$ ) akan dipisahkan berdasarkan 2 buah Garis-Pembagi baru (*LineL* & *LineR*) berdasarkan nilai determinan dari 2 buah titik ekstrim ( $p_1$  dan  $p_2$ ) serta sebuah titik uji ( $p_3$ ) pada himpunan  $S_n$ . Titik ekstrim  $p_1$  dan  $p_2$  adalah titik awal dan titik akhir dari sebuah Garis-Pembagi baru (*LineL* atau *LineR*). Akan terdapat 2 himpunan baru:

- a)  $\det > 0$  terhadap *LineL*, maka titik uji ( $p_3$ ) berada di sebelah kiri garis  $p_1 p_2$   $\rightarrow$  Himpunan  $S_{n1}$
- b)  $\det > 0$  terhadap *LineR*, maka titik uji ( $p_3$ ) berada di sebelah kiri garis  $p_1 p_2$   $\rightarrow$  Himpunan  $S_{n2}$

Untuk **SETIAP** bagian himpunan yang baru, akan ada 2 kemungkinan:

- a) Himpunan tersebut kosong, maka Garis-Pembagi himpunan adalah convex pada bagian himpunan tersebut. Lanjut ke langkah 5.

b) Himpunan tersebut tidak kosong, lakukan langkah **4a.** s.d. langkah **4c.** secara rekursif.

5. Jika **SETIAP** himpunan pada level yang sama telah ditemukan convexnya, maka **COMBINE** convex-convex dari himpunan yang **Se-Level** tersebut. Lanjut ke langkah 6

Jika belum, cari convex dari himpunan teman dengan melakukan langkah **4a.** s.d langkah **4c.**

6. Setiap Convex hasil **COMBINE** pada level **( $n+1$ )** di**COMBINE** lagi untuk convex pada level  **$n$ .** Lanjut ke langkah **7.**

7. Jika belum semua bagian himpunan ditemukan convexnya, maka lakukan langkah **4.** untuk himpunan yang belum memiliki convex.

Jika sudah, maka hasil **COMBINE** akhir merupakan convex hull dari **himpunan awal** ( $S$ )

## B. SOURCE CODE

### 1) Sorting.py

Sorting.py X

src > Sorting.py > quickSort

```
1  ### Python program for implementation of Quicksort Sort
2
3  ''' Proses : - Diambil titik terakhir sebagai pivot di setiap itetasi
4                  - Lalu letakkan pivot tersebut secara terurut di array hasil
5                  dengan meletakkan nilai yang lebih kecil di sebelah kirinya
6                  dan nilai yang lebih besar di sebelah kananya '''
7
8  def partition(arr, low, high):
9      ''' 1.1. Tentukan elemen pivot terlebih dahulu'''
10     i = (low-1)          # i adalah indeks dari elemen terkecil dalam array, "asumsi awal" : elemen terakhir
11     pivotX = arr[high, 0] # penentuan pivot : elemen terakhir
12     pivotY = arr[high, 1]
13
14     ''' 1.2 Mencari semua elemen yang <= dari pivot + aturan
15                 Aturan : Jika nilai x sama, maka bandingkan berdasarkan nilai y'''
16     for j in range(low, high):
17         if (arr[j, 0] < pivotX) or (arr[j, 0] == pivotX and arr[j, 1] <= pivotY):
18             ''' 1.3 Lakukan proses "SWAP" {mengganti elemen} '''
19             i = i+1
20             arr[i, 0], arr[j, 0] = arr[j, 0], arr[i, 0]
21             arr[i, 1], arr[j, 1] = arr[j, 1], arr[i, 1]
22     ''' 1.4 Swap elemen terakhir ditemukan nilai terkecil, dengan elemen terakhir '''
23     arr[i+1, 0], arr[high, 0] = arr[high, 0], arr[i+1, 0]
24     arr[i+1, 1], arr[high, 1] = arr[high, 1], arr[i+1, 1]
25
26     return (i+1)
27
28 # The main function that implements QuickSort
29 # arr[] --> Array to be sorted,
30 # low --> Starting index,
31 # high --> Ending index
32
33 # Function to do Quick sort
34
35
36 def quickSort(arr, low, high):
37     if len(arr) == 1:
38         return arr
39     if low < high:
40
41         # pi is partitioning index, arr[p] is now
42         # at right place
43         pi = partition(arr, low, high)
44
45         # Separately sort elements before
46         # partition and after partition
47         quickSort(arr, low, pi-1)
48         quickSort(arr, pi+1, high)
49
50
```

2) ConvexHull\_v2.py

ConvexHull\_v2.py X

src > ConvexHull\_v2.py > ...

Gleen, 3 hours ago | 1 author (Gleen)

Gleen, 7 hours ago • Overwrite Main

```
1 import numpy as np
2 from Sorting import quickSort
3
4 def DetFrom3Point(p1,p2,p3):
5     mat = np.array([[1,1,1],
6                     [p1[0],p2[0],p3[0]],
7                     [p1[1],p2[1],p3[1]]])
8     return np.linalg.det(mat)
9
10 def AngleFrom3Point(p1,p2,p3):
11     ''' Fungsi: Menerima 2 buah titik (p1 dan p2) yang membentuk segmen garis p1p2 dan mengembalikan sudut p3p1p2
12     |     Prekondisi : p1,p2,p3 adalah np.array 2 dimensi '''
13     ba = p3 - p1
14     bc = p2 - p1
15     cosine = np.dot(ba, bc) / (np.linalg.norm(ba) * np.linalg.norm(bc))
16     return np.arccos(cosine)
17
18 def DistanceFromLine(p1,p2,p3):
19     ''' Fungsi: Menerima 2 buah titik (p1 dan p2) yang membentuk segmen garis dan mengembalikan jarak dari p3 ke garis tsb
20     |     Prekondisi : p1,p2,p3 adalah np.array 2 dimensi '''
21     abs = np.abs
22     cross = np.cross
23     norm = np.linalg.norm
24     return abs(norm(cross(p2-p1, p1-p3)))/norm(p2-p1)
25
26 def NextSimplices(bucket,line,arr): # UNTUK TAHAP KE-2 dst..
27     ''' Proses: - Menerima himpunan titik pada suatu daerah yang ditandai oleh elemen array "arr" sebagai "indeks" dari array "bucket"
28     |     - Array "arr" tidak kosong
29     |     - Array "bucket" memiliki elemen berupa titik secara universal
30     |     - Array "line" memiliki elemen berupa "indeks" dari array "bucket" yang merupakan garis batas daerah sebelumnya
31     |     - Fungsi akan mengembalikan simplex baru dari daerah tersebut
32     |     Prekondisi : Daerah tersebut tidak kosong, minimal elemen array "arr" ada 1 '''
33
34     # a. Titik awal pada garis
35     p1 = bucket[line[0]]
36     # b. Titik akhir pada garis
37     p2 = bucket[line[1]]
38     ''' 1. Cari titik terjauh (p3) sehingga membentuk 2 buah line baru
39     |     lineL adalah p1p3 berupa garis/ simplex sebelah KIRI
40     |     lineR adalah p3p2 berupa garis/ simplex sebelah KANAN '''
41     d = -1 # Jarak tidak mungkin negatif, inisiasi awal sebagai pembanding
42     idx = -1 # Indeks ditemukannya titik dengan jarak terjauh
43     for i in arr:
44         p3 = bucket[i]
45         dTemp = DistanceFromLine(p1,p2,p3)
46         if (dTemp > d):
47             d = dTemp
48             idx = i
49         elif (dTemp == d): # Jika jaraknya sama, makan bandingkan sudut
50             pD = bucket[idx]
51             if (AngleFrom3Point(p1,p2,p3) > AngleFrom3Point(p1,p2,pD)):
52                 d = dTemp
53                 idx = i
54
55     ''' 2. Setelah didapat titik terjauh, bentuk 2 buah line baru '''
56     ### GARIS BAGIAN KIRI
57     lineL = [line[0],idx]
58     p1L = bucket[line[0]]
59     p2L = bucket[idx]
60
61     ### GARIS BAGIAN KANAN
62     lineR = [idx,line[1]]
63     p1R = bucket[idx]
64     p2R = bucket[line[1]]
```

```

65
66     ''' 3. Bagi kumpulan titik pada array "arr" menjadi 2 partisi, namun yg dipakai hanya partisi KIRI
67     | Apabila berada pada garis p1X-p2X, maka abaikan '''
68     newSimplices = []
69     arrL = []
70     arrR = []
71     ''' 3.1. Check di garis atau tidak untuk kedua garis batas baru, di garis (det=0) '''
72     for i in arr:
73         ''' Note: pMax tidak perlu dicek karena merupakan titik pembentuk garis'''
74         if i==idx:
75             continue
76         p3 = bucket[i]
77         ''' 3.1.a. Check untuk bagian kiri lineL,
78             3.1.b. Check untuk bagian kiri lineR,
79             apabila (det>0) maka masukkan ke array selanjutnya untuk diperiksa '''
80         detL = DetFrom3Point(p1L,p2L,p3)
81         detR = DetFrom3Point(p1R,p2R,p3)
82         if (detL > 0):
83             arrL += [i]
84         elif (detR > 0):
85             arrR += [i]
86
87     ''' 3.4 CONQUER & COMBINE masing-masing sisi (INI TAHAP TERSULIT)'''
88     # URUTAN CONQUER = KIRI -> KANAN
89     if arrL==[] and arrR==[]:
90         newSimplices += [lineL]
91         newSimplices += [lineR]
92     elif arrL==[] and arrR!=[]:
93         newSimplices += [lineL]
94         newSimplices += NextSimplices(bucket,lineR,arrR)
95     elif arrR==[] and arrL!=[]:
96         newSimplices += [lineR]
97         newSimplices += NextSimplices(bucket,lineL,arrL)
98     else: # {left!=[] and right!=[]}
99         newSimplices += NextSimplices(bucket,lineL,arrL)
100         newSimplices += NextSimplices(bucket,lineR,arrR)
101
102     return newSimplices
103

```

```

104 104 ▼ def ConvexHull_v2(bucket):
105     ''' 1. Diurutkan berdasarkan nilai x dan y '''
106     quickSort(bucket,0,len(bucket)-1)
107
108     ''' 2. Buat p1-pn sebagai line batas awal '''
109     line = [0,len(bucket)-1]
110     # a. Titik awal pada garis
111     p1 = bucket[line[0]]
112     # b. Titik akhir pada garis
113     p2 = bucket[line[1]]
114
115 115 ▼
116     ''' 3. Bagi kumpulan titik yang lain menjadi 2 partisi
117     | Apabila berada pada garis p1-pn, maka abaikan '''
117     simplices = []
118     left = []
119     right = []
120     ''' 3.1. Check di garis atau tidak, di garis (det=0) '''
121 121 ▼
122     for i in range(1,len(bucket)-1):
123         p3 = bucket[i]
124         det = DetFrom3Point(p1,p2,p3)
125         ''' 3.2 Jika tidak digaris, bagi menjadi bagian KIRI (det>0) dan KANAN (det<0) '''
126 125 ▼
127         if (det > 0):
128             left += [i]
129         elif (det < 0):
130             right += [i]
131
132     ''' 3.3 Lakukan pengecekan secara rekursif untuk bagian KIRI dan bagian KANAN'''
133     lineL = [line[0],line[1]]
134     lineR = [line[1],line[0]]
135     simplices += NextSimplices(bucket,lineL,left)
136     simplices += NextSimplices(bucket,lineR,right)
137
138     return simplices

```

3) Main.ipynb

Main.ipynb X

src > Main.ipynb > import pandas as pdfrom sklearn import datasetsfrom matplotlib.pyplot as pltfrom ConvexHull\_v2 import ConvexHull\_v2def printPilihan():print(" 1. Sepal Width (y) vs Sepal Length (x) :: IRIS-Datasets")print(" 2. Petal Width (y) vs Petal Length (x) :: IRIS-Datasets")print(" 3. Color Intensity (y) vs Alcohol % (x) :: WINE-Datasets")print(" 4. Flavanoids (y) vs Total Phenols (x) :: WINE-Datasets")print(" 5. Mean Texture (y) vs Mean Radius (x) :: Breas\_Cancer-Datasets")print(" 6. Worst Texture (y) vs Worst Radius (x) :: Breas\_Cancer-Datasets")pil = int(input("Masukkan pilihan : "))return pil

+ Code+ Markdown▶ Run AllClear Outputs of All CellsRestartInterruptVariablesOutline

```
import pandas as pd
from sklearn import datasets
import matplotlib.pyplot as plt
from ConvexHull_v2 import ConvexHull_v2

def printPilihan():
    print("1. Sepal Width (y) vs Sepal Length (x) :: IRIS-Datasets")
    print("2. Petal Width (y) vs Petal Length (x) :: IRIS-Datasets")
    print("3. Color Intensity (y) vs Alcohol % (x) :: WINE-Datasets")
    print("4. Flavanoids (y) vs Total Phenols (x) :: WINE-Datasets")
    print("5. Mean Texture (y) vs Mean Radius (x) :: Breas_Cancer-Datasets")
    print("6. Worst Texture (y) vs Worst Radius (x) :: Breas_Cancer-Datasets")
    pil = int(input("Masukkan pilihan : "))
    return pil

''' ..... '''

### USER INTERFACE
print("Selamat datang di Uji Coba Convex Hull v.2.0.1")
print("-----")
print("Terdapat beberapa pilihan visualisasi :)")
pil = printPilihan()

while not(1<=pil<=6):
    print("Masukan anda tidak sesuai...! Harap masukkan pilihan kembali...!")
    printPilihan()

print("Pilihan anda adalah :",pil)

if (pil==1 or pil==2):
    data = datasets.load_iris()
    if pil==1:
        title = 'Sepal Width vs Sepal Length'
        col1 = 0
        col2 = 1
    elif pil==2:
        title = 'Petal Width vs Petal Length'
        col1 = 2
        col2 = 3
    elif (pil==3 or pil==4):
        data = datasets.load_wine()
        if pil==3:
            title = 'Color Intensity vs alcohol'
            col1 = 0
            col2 = 9
        elif pil==4:
            title = 'Flavanoids vs Total Phenols'
            col1 = 5
            col2 = 6
    elif (pil==5 or pil==6):
        data = datasets.load_breast_cancer()
        if pil==5:
            title = 'Mean Texture vs Mean Radius'
            col1 = 0
            col2 = 1
        elif pil==6:
            title = 'Worst Texture vs Worst Radius'
            col1 = 20
            col2 = 21

#create a DataFrame
df = pd.DataFrame(data.data, columns=data.feature_names)
df['Target'] = pd.DataFrame(data.target)
print()
print("Berikut Contoh Data yang Diolah :)")
df.head()

[2] ✓ 0.9s
```

```
# Visualizing Datasets
colors = ['b','r','g']
fig = plt.figure(figsize = (10, 6))
plt.title(title)
plt.xlabel(data.feature_names[col1])
plt.ylabel(data.feature_names[col2])

print("Berikut Visualisasi Data yang Diolah :)")
for i in range(len(data.target_names)):
    each = df[df['Target'] == i]          # untuk mengambil data dengan target yang seragam
    bucket = each.iloc[:,[col1,col2]].values # untuk mendapatkan array berupa posisi x dan y, berukuran n x 2, n adalah banyak data
    simplices = ConvexHull_v2(bucket)      # bagian ini diganti dengan hasil implementasi ConvexHull Divide & Conquer
    plt.scatter(bucket[:, 0], bucket[:, 1], label=data.target_names[i])
    for simplex in simplices:
        plt.plot(bucket[simplex, 0], bucket[simplex, 1], colors[i])
plt.legend()

# Tampilkan dan simpan hasil visualisasi
rect = fig.patch
rect.set_facecolor("white")
plt.savefig("../output/" + str(pil) + ". " + title + ".png")
plt.show()
```

[3] ✓ 0.5s

C. CONTOH EKSEKUSI PROGRAM

1. Sepal Width (y) vs Sepal Length (x) :: IRIS-Datasets

...

Selamat datang di Uji Coba Convex Hull v.2.0.1

Terdapat beberapa pilihan visualisasi :

1. Sepal Width (y) vs Sepal Length (x) :: IRIS-Datasets

2. Petal Width (y) vs Petal Length (x) :: IRIS-Datasets

3. Color Intensity (y) vs Alcohol % (x) :: WINE-Datasets

4. Flavanoids (y) vs Total Phenols (x) :: WINE-Datasets

5. Mean Texture (y) vs Mean Radius (x) :: Breas\_Cancer-Datasets

6. Worst Texture (y) vs Worst Radius (x) :: Breas\_Cancer-Datasets

Pilihan anda adalah : 1

...

Berikut Contoh Data yang Diolah :

</>

|   | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | Target |
|---|-------------------|------------------|-------------------|------------------|--------|
| 0 | 5.1               | 3.5              | 1.4               | 0.2              | 0      |
| 1 | 4.9               | 3.0              | 1.4               | 0.2              | 0      |
| 2 | 4.7               | 3.2              | 1.3               | 0.2              | 0      |
| 3 | 4.6               | 3.1              | 1.5               | 0.2              | 0      |
| 4 | 5.0               | 3.6              | 1.4               | 0.2              | 0      |

...

Berikut Visualisasi Data yang Diolah :

</>



2. Petal Width (y) vs Petal Length (x) :: IRIS-Datasets

```
... Selamat datang di Uji Coba Convex Hull v.2.0.1
=====

Terdapat beberapa pilihan visualisasi :

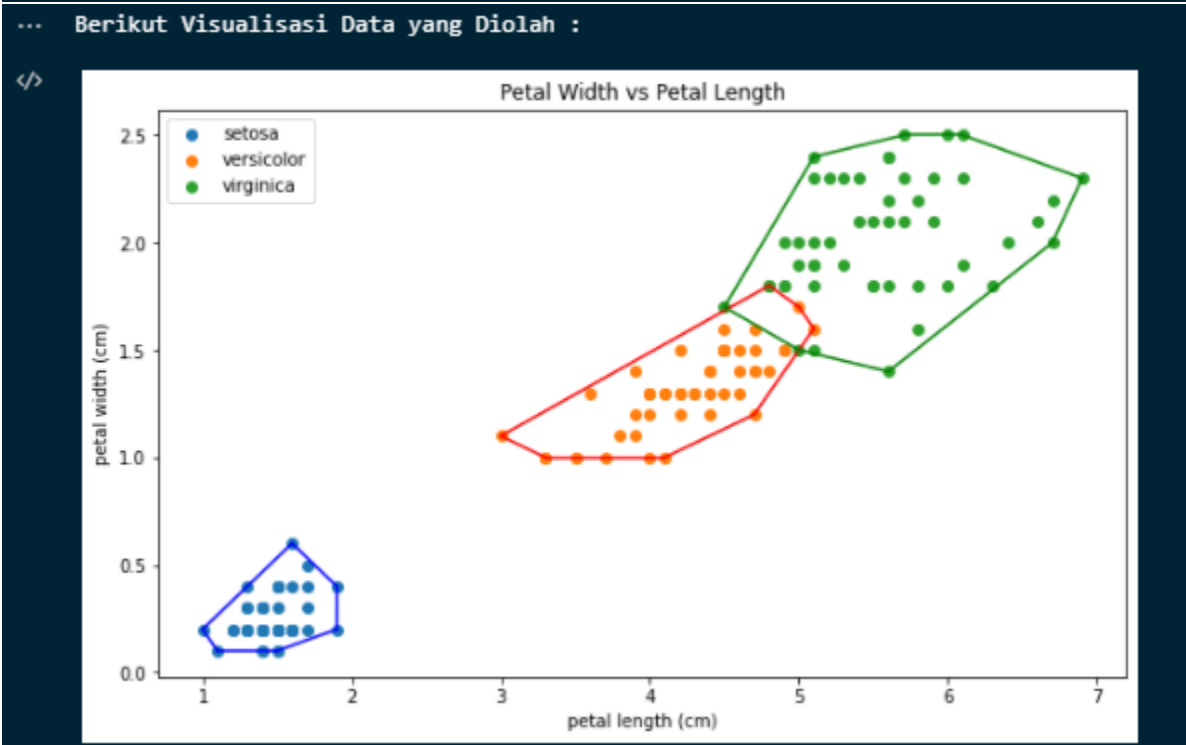
1. Sepal Width (y) vs Sepal Length (x)    :: IRIS-Datasets
2. Petal Width (y) vs Petal Length (x)    :: IRIS-Datasets
3. Color Intensity (y) vs Alcohol % (x)   :: WINE-Datasets
4. Flavanoids (y) vs Total Phenols (x)    :: WINE-Datasets
5. Mean Texture (y) vs Mean Radius (x)    :: Breas_Cancer-Datasets
6. Worst Texture (y) vs Worst Radius (x)  :: Breas_Cancer-Datasets

Pilihan anda adalah : 2
```

```
... Berikut Contoh Data yang Diolah :
```

```
</>
```

|   | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | Target |
|---|-------------------|------------------|-------------------|------------------|--------|
| 0 | 5.1               | 3.5              | 1.4               | 0.2              | 0      |
| 1 | 4.9               | 3.0              | 1.4               | 0.2              | 0      |
| 2 | 4.7               | 3.2              | 1.3               | 0.2              | 0      |
| 3 | 4.6               | 3.1              | 1.5               | 0.2              | 0      |
| 4 | 5.0               | 3.6              | 1.4               | 0.2              | 0      |



3. Color Intensity (y) vs Alcohol % (x) :: WINE-Datasets

```
... Selamat datang di Uji Coba Convex Hull v.2.0.1

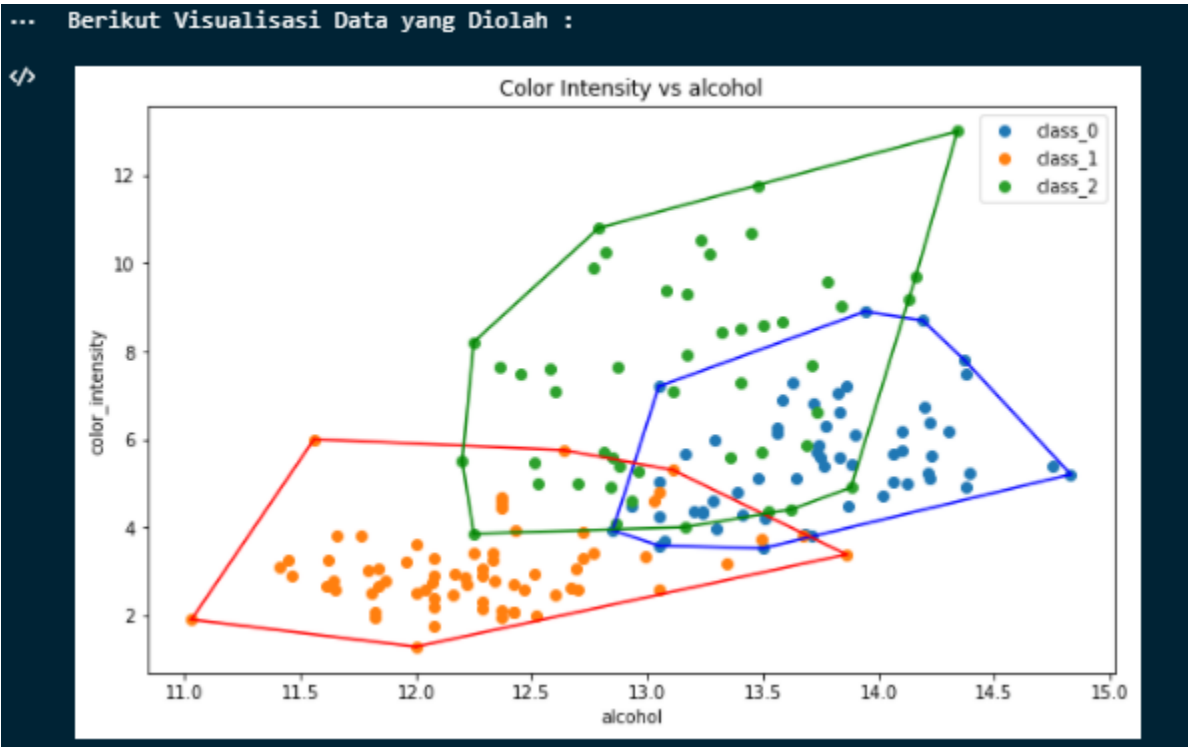
Terdapat beberapa pilihan visualisasi :

1. Sepal Width (y) vs Sepal Length (x) :: IRIS-Datasets
2. Petal Width (y) vs Petal Length (x) :: IRIS-Datasets
3. Color Intensity (y) vs Alcohol % (x) :: WINE-Datasets
4. Flavanoids (y) vs Total Phenols (x) :: WINE-Datasets
5. Mean Texture (y) vs Mean Radius (x) :: Breas_Cancer-Datasets
6. Worst Texture (y) vs Worst Radius (x) :: Breas_Cancer-Datasets

Pilihan anda adalah : 3
```

... Berikut Contoh Data yang Diolah :

| </> | alcohol | malic_acid | ash  | alkalinity_of_ash | magnesium | total_phenols | flavanoids | nonflavanoid_phenols | proanthocyanins | color_intensity | hue  | od280/od315_of_diluted_wines | proline | Target |
|-----|---------|------------|------|-------------------|-----------|---------------|------------|----------------------|-----------------|-----------------|------|------------------------------|---------|--------|
| 0   | 14.23   | 1.71       | 2.43 | 15.6              | 127.0     | 2.80          | 3.06       | 0.28                 | 2.29            | 5.64            | 1.04 | 3.92                         | 1065.0  | 0      |
| 1   | 13.20   | 1.78       | 2.14 | 11.2              | 100.0     | 2.65          | 2.76       | 0.26                 | 1.28            | 4.38            | 1.05 | 3.40                         | 1050.0  | 0      |
| 2   | 13.16   | 2.36       | 2.67 | 18.6              | 101.0     | 2.80          | 3.24       | 0.30                 | 2.81            | 5.68            | 1.03 | 3.17                         | 1185.0  | 0      |
| 3   | 14.37   | 1.95       | 2.50 | 16.8              | 113.0     | 3.85          | 3.49       | 0.24                 | 2.18            | 7.80            | 0.86 | 3.45                         | 1480.0  | 0      |
| 4   | 13.24   | 2.59       | 2.87 | 21.0              | 118.0     | 2.80          | 2.69       | 0.39                 | 1.82            | 4.32            | 1.04 | 2.93                         | 735.0   | 0      |



4. Flavanoids (y) vs Total Phenols (x) :: WINE-Datasets

```
... Selamat datang di Uji Coba Convex Hull v.2.0.1

~~~~~

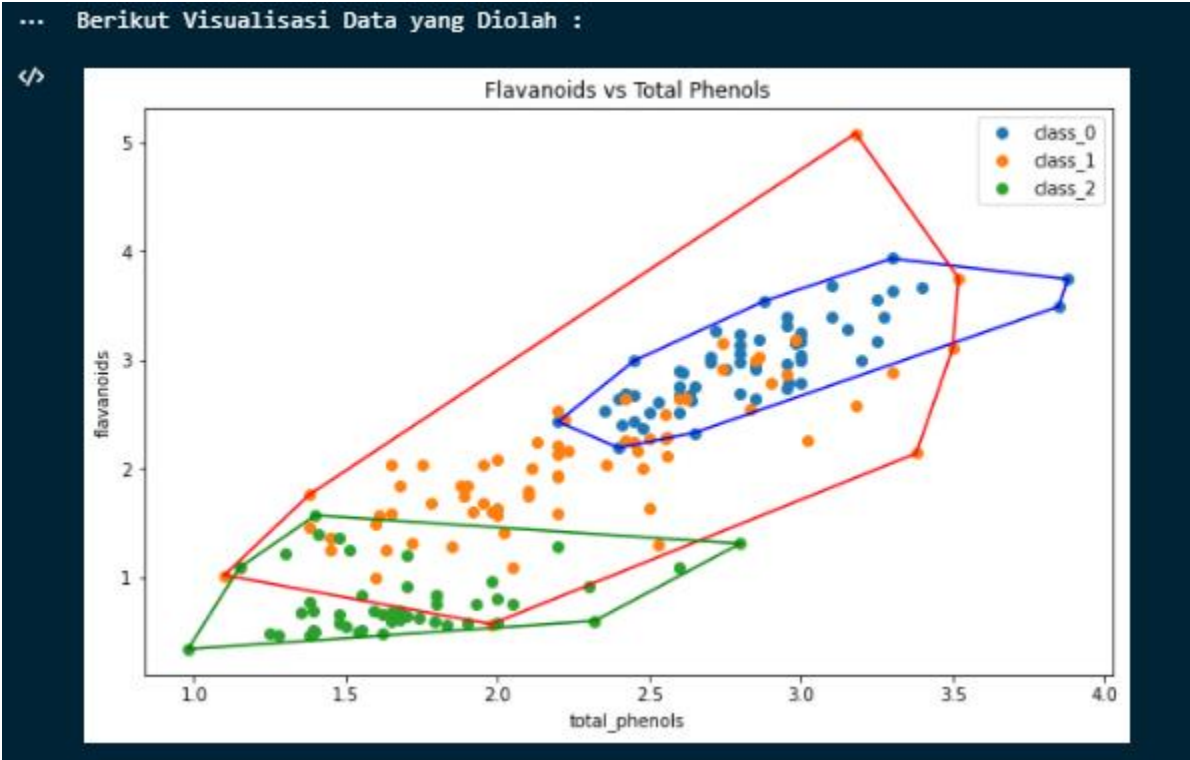
Terdapat beberapa pilihan visualisasi :

1. Sepal Width (y) vs Sepal Length (x) :: IRIS-Datasets
2. Petal Width (y) vs Petal Length (x) :: IRIS-Datasets
3. Color Intensity (y) vs Alcohol % (x) :: WINE-Datasets
4. Flavanoids (y) vs Total Phenols (x) :: WINE-Datasets
5. Mean Texture (y) vs Mean Radius (x) :: Breas_Cancer-Datasets
6. Worst Texture (y) vs Worst Radius (x) :: Breas_Cancer-Datasets

Pilihan anda adalah : 4
```

Berikut Contoh Data yang Diolah :

|   | alcohol | malic_acid | ash  | alkalinity_of_ash | magnesium | total_phenols | flavanoids | nonflavanoid_phenols | proanthocyanins | color_intensity | hue  | od280/od315_of_diluted_wines | proline | Target |
|---|---------|------------|------|-------------------|-----------|---------------|------------|----------------------|-----------------|-----------------|------|------------------------------|---------|--------|
| 0 | 14.23   | 1.71       | 2.43 | 15.6              | 127.0     | 2.80          | 3.06       | 0.28                 | 2.29            | 5.64            | 1.04 | 3.92                         | 1065.0  | 0      |
| 1 | 13.20   | 1.78       | 2.14 | 11.2              | 100.0     | 2.65          | 2.76       | 0.26                 | 1.28            | 4.38            | 1.05 | 3.40                         | 1050.0  | 0      |
| 2 | 13.16   | 2.36       | 2.67 | 18.6              | 101.0     | 2.80          | 3.24       | 0.30                 | 2.81            | 5.68            | 1.03 | 3.17                         | 1185.0  | 0      |
| 3 | 14.37   | 1.95       | 2.50 | 16.8              | 113.0     | 3.85          | 3.49       | 0.24                 | 2.18            | 7.80            | 0.86 | 3.45                         | 1480.0  | 0      |
| 4 | 13.24   | 2.59       | 2.87 | 21.0              | 118.0     | 2.80          | 2.69       | 0.39                 | 1.82            | 4.32            | 1.04 | 2.93                         | 735.0   | 0      |



5. Mean Texture (y) vs Mean Radius (x) :: Breas\_Cancer-Datasets

```
... Selamat datang di Uji Coba Convex Hull v.2.0.1

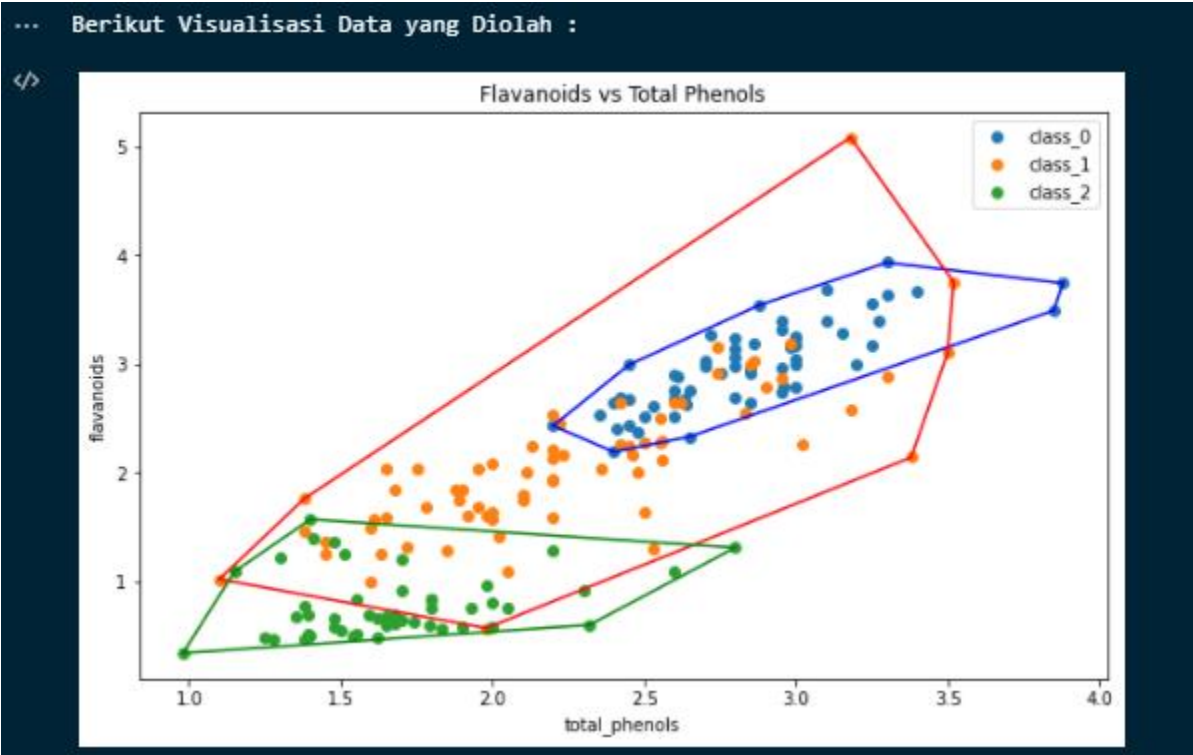
Terdapat beberapa pilihan visualisasi :

1. Sepal Width (y) vs Sepal Length (x) :: IRIS-Datasets
2. Petal Width (y) vs Petal Length (x) :: IRIS-Datasets
3. Color Intensity (y) vs Alcohol % (x) :: WINE-Datasets
4. Flavanoids (y) vs Total Phenols (x) :: WINE-Datasets
5. Mean Texture (y) vs Mean Radius (x) :: Breas_Cancer-Datasets
6. Worst Texture (y) vs Worst Radius (x) :: Breas_Cancer-Datasets

Pilihan anda adalah : 4
```

... Berikut Contoh Data yang Diolah :

|   | alcohol | malic_acid | ash  | alkalinity_of_ash | magnesium | total_phenols | flavanoids | nonflavanoid_phenols | proanthocyanins | color_intensity | hue  | od280/od315_of_diluted_wines | proline | Target |
|---|---------|------------|------|-------------------|-----------|---------------|------------|----------------------|-----------------|-----------------|------|------------------------------|---------|--------|
| 0 | 14.23   | 1.71       | 2.43 | 15.6              | 127.0     | 2.80          | 3.06       | 0.28                 | 2.29            | 5.64            | 1.04 | 3.92                         | 1065.0  | 0      |
| 1 | 13.20   | 1.78       | 2.14 | 11.2              | 100.0     | 2.65          | 2.76       | 0.26                 | 1.28            | 4.38            | 1.05 | 3.40                         | 1050.0  | 0      |
| 2 | 13.16   | 2.36       | 2.67 | 18.6              | 101.0     | 2.80          | 3.24       | 0.30                 | 2.81            | 5.68            | 1.03 | 3.17                         | 1185.0  | 0      |
| 3 | 14.37   | 1.95       | 2.50 | 16.8              | 113.0     | 3.85          | 3.49       | 0.24                 | 2.18            | 7.80            | 0.86 | 3.45                         | 1480.0  | 0      |
| 4 | 13.24   | 2.59       | 2.87 | 21.0              | 118.0     | 2.80          | 2.69       | 0.39                 | 1.82            | 4.32            | 1.04 | 2.93                         | 735.0   | 0      |



6. Worst Texture (y) vs Worst Radius (x) :: Breas\_Cancer-Datasets

```
... Selamat datang di Uji Coba Convex Hull v.2.0.1

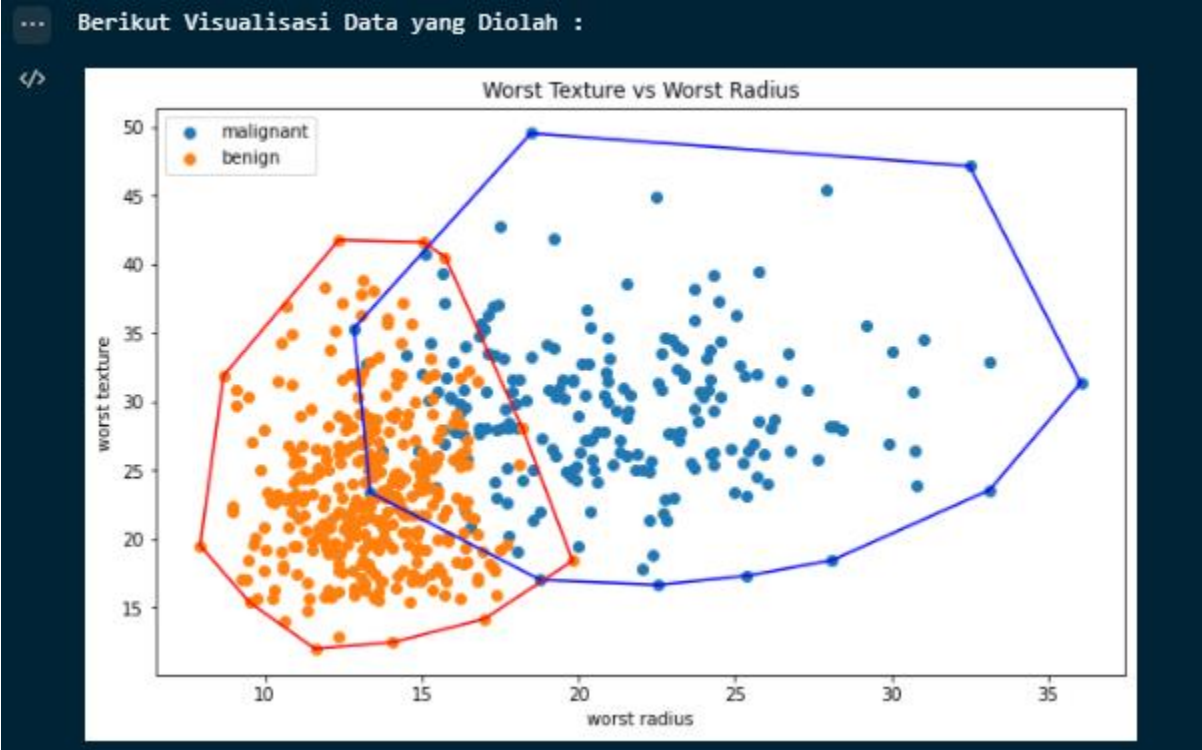
Terdapat beberapa pilihan visualisasi :

1. Sepal Width (y) vs Sepal Length (x)      :: IRIS-Datasets
2. Petal Width (y) vs Petal Length (x)      :: IRIS-Datasets
3. Color Intensity (y) vs Alcohol % (x)      :: WINE-Datasets
4. Flavanoids (y) vs Total Phenols (x)       :: WINE-Datasets
5. Mean Texture (y) vs Mean Radius (x)       :: Breas_Cancer-Datasets
6. Worst Texture (y) vs Worst Radius (x)     :: Breas_Cancer-Datasets

Pilihan anda adalah : 6
```

... Berikut Contoh Data yang Diolah :

|   | alcohol | malic acid | ash  | alcalinity_of_ash | magnesium | total_phenols | flavanoids | nonflavanoid_phenols | proanthocyanins | color_intensity | hue  | od280/od315_of_diluted_wines | proline | Target |
|---|---------|------------|------|-------------------|-----------|---------------|------------|----------------------|-----------------|-----------------|------|------------------------------|---------|--------|
| 0 | 14.23   | 1.71       | 2.43 | 15.6              | 127.0     | 2.80          | 3.06       | 0.28                 | 2.29            | 5.64            | 1.04 | 3.92                         | 1065.0  | 0      |
| 1 | 13.20   | 1.78       | 2.14 | 11.2              | 100.0     | 2.65          | 2.76       | 0.26                 | 1.28            | 4.38            | 1.05 | 3.40                         | 1050.0  | 0      |
| 2 | 13.16   | 2.36       | 2.67 | 18.6              | 101.0     | 2.80          | 3.24       | 0.30                 | 2.81            | 5.68            | 1.03 | 3.17                         | 1185.0  | 0      |
| 3 | 14.37   | 1.95       | 2.50 | 16.8              | 113.0     | 3.85          | 3.49       | 0.24                 | 2.18            | 7.80            | 0.86 | 3.45                         | 1480.0  | 0      |
| 4 | 13.24   | 2.59       | 2.87 | 21.0              | 118.0     | 2.80          | 2.69       | 0.39                 | 1.82            | 4.32            | 1.04 | 2.93                         | 735.0   | 0      |



D. LINK

Source code dan laporan dapat diakses melalui:  
[https://github.com/Amike31/LINIER\\_SEPARABILITY\\_TEST](https://github.com/Amike31/LINIER_SEPARABILITY_TEST)

| Poin  | Ya | Tidak |
|---|----|-------|
| 1. Pustaka <i>myConvexHull</i> berhasil dibuat dan tidak ada kesalahan  | √  |       |
| 2. <i>Convex hull</i> yang dihasilkan 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.                                  | √  |       |