

# Laporan Tugas Kecil 2 IF2211 Strategi Algoritma

13520152 – Muhammad Fahmi Irfan

## Bab 1 : Deskripsi Singkat Program

Program MyConvexHull ini merupakan program yang dapat menentukan *convex hull* dari kumpulan data dua dimensi. Program ini dibuat menggunakan Bahasa Pemrograman Python dan menggunakan Algoritma Divide and Conquer.

## Bab 2 : Algoritma Divide and Conquer

Algoritma Divide and Conquer merupakan algoritma yang memecah persoalan menjadi persoalan-persolan kecil, lalu menyelesaikan persoalan-persolan kecil tersebut, kemudian solusi-solusi tersebut digabung menjadi solusi persoalan yang besar. Secara garis besar, terdapat tiga bagian algoritma ini, yaitu Divide (membagi persoalan menjadi beberapa upa-persoalan yang memiliki kemiripan dengan persoalan semula namun ukurannya lebih kecil), Conquer (menyelesaikan upa-persoalan tersebut dan menghasilkan upa-solusi), dan Combine (menggabungkan upa-solusi menjadi solusi keseluruhan).

Dalam program ini, ide algoritma yang digunakan ialah sebagai berikut.

1. Mencari titik paling kiri (misalkan A) dan paling kanan (misalkan B) dari suatu kumpulan titik dua dimensi. Jika ada dua titik paling kiri dengan absis yang sama, pilih titik yang ordinatnya lebih kecil. Jika ada dua titik paling kanan dengan absis yang sama, pilih titik yang ordinatnya lebih besar.
2. Membagi titik-titik menjadi dua bagian, yaitu bagian di atas garis AB (misalkan S1) dan bagian di bawah garis AB (misalkan S2).
3. Algoritma QuickHull(S,A,B) yang digunakan ialah sebagai berikut
  - a. Jika S kosong, buat garis AB
  - b. Jika S tidak kosong, cari titik terjauh dari garis AB dengan membandingkan luas dari segitiga ABP, dengan P merupakan titik pada S. Luas terbesar akan menghasilkan jarak terjauh (karena alas dari segitiga dari semua segitiga ABP yang mungkin sama, yaitu AB, maka perbedaan tinggi segitiga akan menghasilkan perbedaan luas dan tinggi dari segitiga tersebut ialah jarak dari titik P ke garis AB). Jika sama, titik yang dipilih ialah titik yang menghasilkan sudut PAB paling besar. Untuk jarak dari P ke AB yang sama, sudut PAB akan lebih besar jika titik P lebih dekat dengan titik A.Jika S tidak kosong dan sudah ditentukan suatu titik dari tahap ketiga yang dinamakan Pmax, Bagi titik-titik menjadi tiga bagian, yaitu titik-titik di atas garis APmax (misalkan Skiri), titik-titik di atas garis PmaxB (misalkan Skanan), dan titik-titik di dalam segitiga ABPmax. Lakukan lagi QuickHull(Skiri,A,Pmax) dan QuickHull(Skanan,Pmax,B)
4. Lakukan QuickHull(S1,A,B) dan QuickHull(S2,B,A). Pada bagian bawah, titik-titik akan dilihat secara terbalik (titik paling kanan ada di kiri, titik paling kiri ada di kanan, Skiri berisi titik-titik yang ada di kanan APmax dan Skanan berisi titik-titik yang ada di kiri PmaxB). Hal ini dilakukan karena perhitungan algoritma QuickHull untuk titik-titik yang ada di bawah akan memiliki perhitungan yang sama dengan titik-titik yang ada di atas jika dilihat dengan prespektif yang terbalik.

## Bab 3 : Kode Program

Program dibagi menjadi dua file, yaitu file *MyConvexHull.py* dan *main.ipynb*. *MyConvexHull.py* berisi algoritma Convex Hull yang digunakan dan *main.ipynb* berisi implementasi dari algoritma tersebut yang mengeluarkan *output* berupa grafik hasil algoritma tersebut.

### a. MyConvexHull.py

Berikut merupakan isi dari file *MyConvexHull.py*

```
zero = 10**-9

def twice_area(p_left, p_right, p):
    # return 2*area of triangle p_left-p_right-p
    # jika nilainya positif, p ada di atas garis p_left-p_right dan sebaliknya
    # jika posisinya ditukar (p_left ada di kanan, p_right ada di kiri),
    # nilainya positif jika dan hanya jika p ada di bawah garis
    # perbandingan luas = perbandingan tinggi jika alasnya sama
    detval = p_left[0]*p_right[1] + p[0]*p_left[1] + p_right[0]*p[1] -
    p[0]*p_right[1] - p_right[0]*p_left[1] - p_left[0]*p[1]
    return detval

def ConvexHull(s):
    # return nx2 array, each row contains two indices of s which will be drawn
    # a line that connects them
    if(len(s) == 0) : return [[]]

    simplices = []
    p_left = 0
    p_right = 0
    for i in range(len(s)):
        if(s[i][0]<s[p_left][0]): p_left = i
        elif(s[i][0]==s[p_left][0] and s[i][1]<s[p_left][1]): p_left = i
        if(s[i][0]>s[p_right][0]): p_right = i
        elif(s[i][0]==s[p_right][0] and s[i][1]>s[p_right][1]): p_right = i

    upper = []
    lower = []
    for i in range(len(s)):
        if(twice_area(s[p_left],s[p_right],s[i])>zero and i!=p_left and i!=
p_right): upper.append(i)
        elif(twice_area(s[p_left],s[p_right],s[i])<zero and i!=p_left and
i!=p_right): lower.append(i)

    simplices = qhull(s,upper,p_left,p_right) + qhull(s,lower,p_right,p_left)
    return simplices

def qhull(s, indices, p_left, p_right):
    # return nx2 array, each row contains two indices of s which will be drawn a
    # line that connects them
    if(len(indices) == 0): return [[p_left, p_right]]
```

```

else:
    farthest_point = indices[0]
    highest_tw_area = twice_area(s[p_left],s[p_right],s[farthest_point])
    for p in indices:
        tw_area = abs(twice_area(s[p_left],s[p_right],s[p]))
        if (tw_area>highest_tw_area or (abs(highest_tw_area - tw_area)<=zero and
abs(s[p][0]-s[p_left][0]) < abs(s[farthest_point][0] - s[p_left][0]))) :
            farthest_point = p
            highest_tw_area = tw_area

    left_points = []
    right_points = []
    for p in indices:
        if(twice_area(s[p_left],s[farthest_point],s[p])>zero and p!=p_left and
p!=farthest_point):
            left_points.append(p)
        elif(twice_area(s[farthest_point],s[p_right],s[p])>zero and p!=p_right
and p!=farthest_point):
            right_points.append(p)

    simplices = qhull(s,left_points,p_left,farthest_point) +
qhull(s,right_points,farthest_point,p_right)

    return simplices

```

## b. Main.ipynb

Berikut merupakan kode untuk membaca dan menggunakan data iris.

```

#Data Iris
import numpy
import pandas as pd
import matplotlib.pyplot as plt
from myConvexHull import ConvexHull
from sklearn import datasets

data = datasets.load_iris()

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

#visualisasi hasil ConvexHull
plt.figure(figsize = (10, 6))
colors = ['b','r','g']
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 s in hull:
        plt.plot(bucket[s,0], bucket[s,1], colors[i])
plt.legend()

plt.figure(figsize = (10, 6))
colors = ['b','r','g']
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 s in hull:
        plt.plot(bucket[s,0], bucket[s,1], colors[i])
plt.legend()

```

Berikut merupakan kode untuk membaca dan menggunakan data wine.

```

#Data Wine
import numpy
import pandas as pd
import matplotlib.pyplot as plt
from myConvexHull import ConvexHull
from sklearn import datasets

data = datasets.load_wine()

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

#visualisasi hasil ConvexHull
plt.figure(figsize = (10, 6))
colors = ['b','r','g']
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 s in hull:
        plt.plot(bucket[s,0], bucket[s,1], colors[i])
plt.legend()

plt.figure(figsize = (10, 6))
colors = ['b','r','g']
plt.title('Ash vs Alkalinity of Ash')
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 s in hull:
        plt.plot(bucket[s,0], bucket[s,1], colors[i])
plt.legend()

plt.figure(figsize = (10, 6))
colors = ['b','r','g']
plt.title('Color Intensity vs Hue')
plt.xlabel(data.feature_names[9])
plt.ylabel(data.feature_names[10])
for i in range(len(data.target_names)):
    bucket = df[df['Target'] == i]
    bucket = bucket.iloc[:,[9,10]].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 s in hull:
        plt.plot(bucket[s,0], bucket[s,1], colors[i])
plt.legend()

```

Berikut merupakan kode untuk membaca dan menggunakan data breast\_cancer

```

#Data Breast Cancer
import numpy
import pandas as pd
import matplotlib.pyplot as plt
from myConvexHull import ConvexHull

from sklearn import datasets
data = datasets.load_breast_cancer()

```

```

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

#visualisasi hasil ConvexHull
plt.figure(figsize = (10,9))
colors = ['blue','orange']
plt.title('Mean Radius vs Mean Texture')
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 s in hull:
        plt.plot(bucket[s,0], bucket[s,1], colors[i])
plt.legend()

plt.figure(figsize = (10,9))
colors = ['blue','orange']
plt.title('Mean Perimeter vs Perimeter Error')
plt.xlabel(data.feature_names[2])
plt.ylabel(data.feature_names[12])
for i in range(len(data.target_names)):
    bucket = df[df['Target'] == i]
    bucket = bucket.iloc[:,[2,12]].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 s in hull:
        plt.plot(bucket[s,0], bucket[s,1], colors[i])
plt.legend()

plt.figure(figsize = (10,9))
colors = ['blue','orange']
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 s in hull:
        plt.plot(bucket[s,0], bucket[s,1], colors[i])

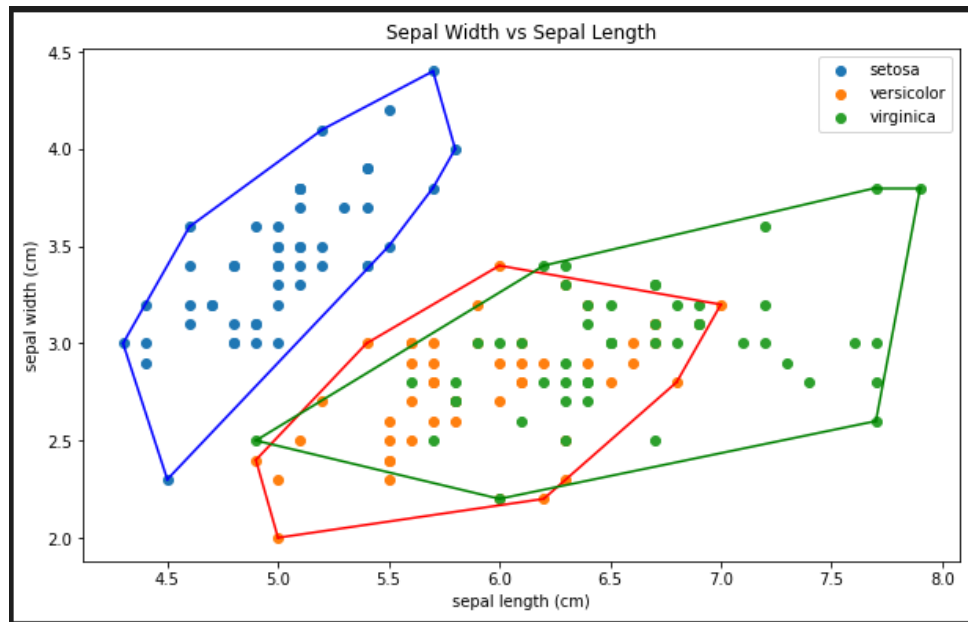
```

```
plt.legend()
```

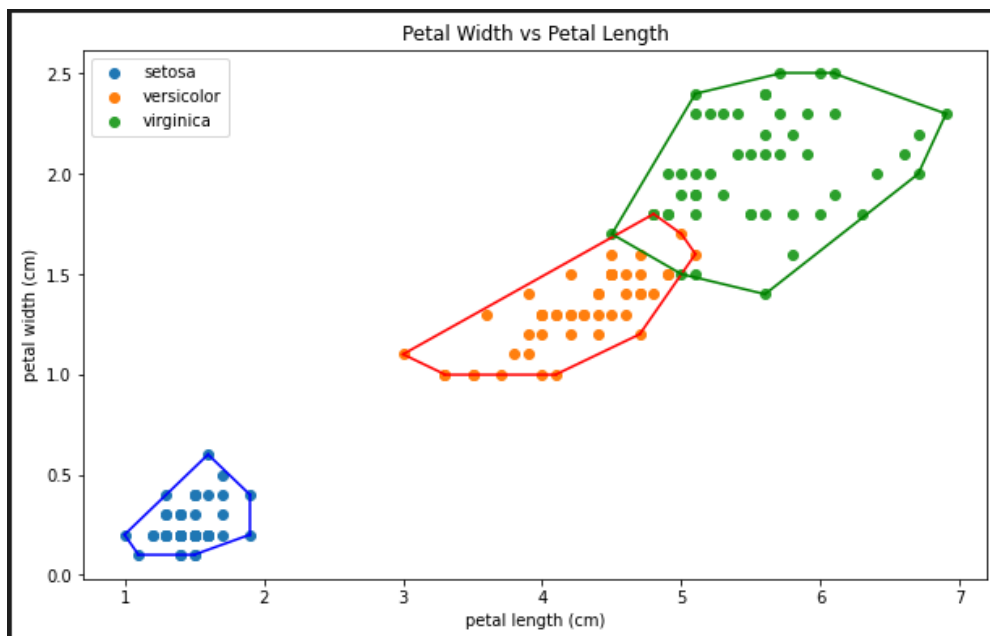
## Bab 4 : Hasil Program

### a. Data Iris

Berikut merupakan luaran program yang menggunakan data iris.



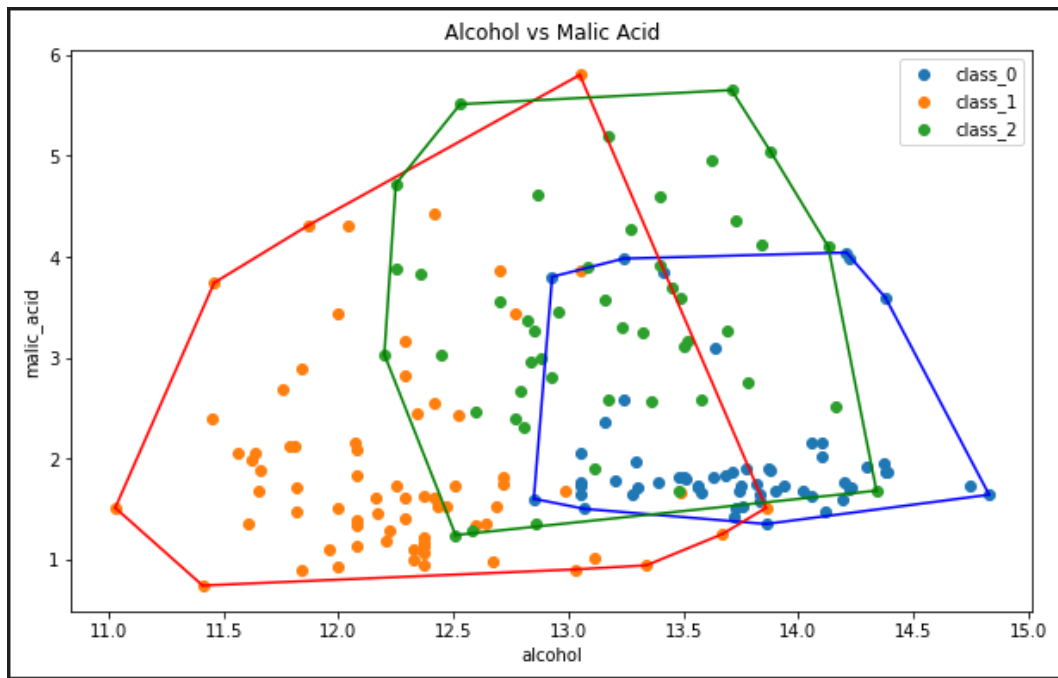
**Gambar 4.1.1.** Luaran program dengan parameter sepal length dan sepal width pada data iris



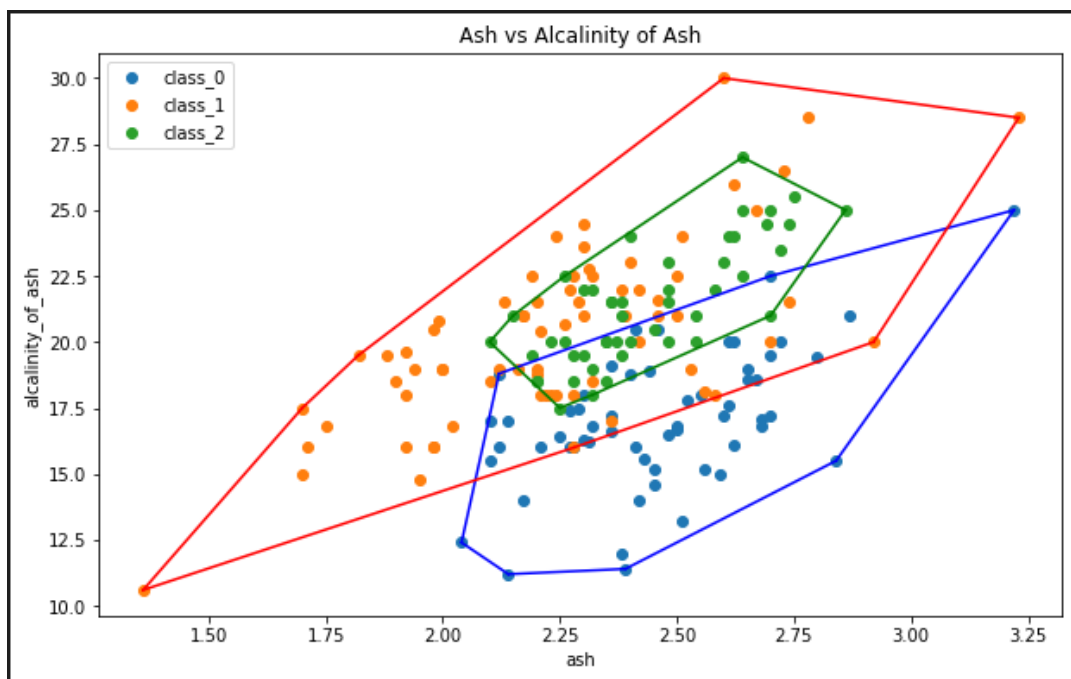
**Gambar 4.1.2.** Luaran program dengan parameter petal length dan petal width pada data iris

### b. Data Wine

Berikut merupakan luaran dari program yang menggunakan data wine.

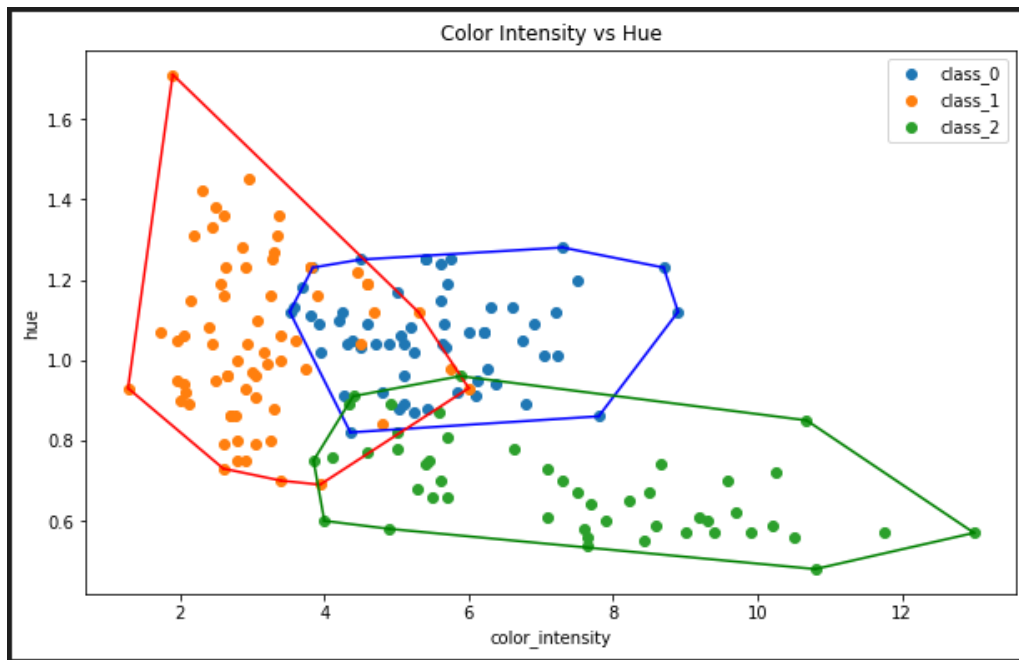


**Gambar 4.2.1.** Luaran program dengan parameter alcohol dan malic\_acid pada data wine



**Gambar 4.2.2.** Luaran program dengan parameter ash dan alkalinity\_of\_ash pada data wine

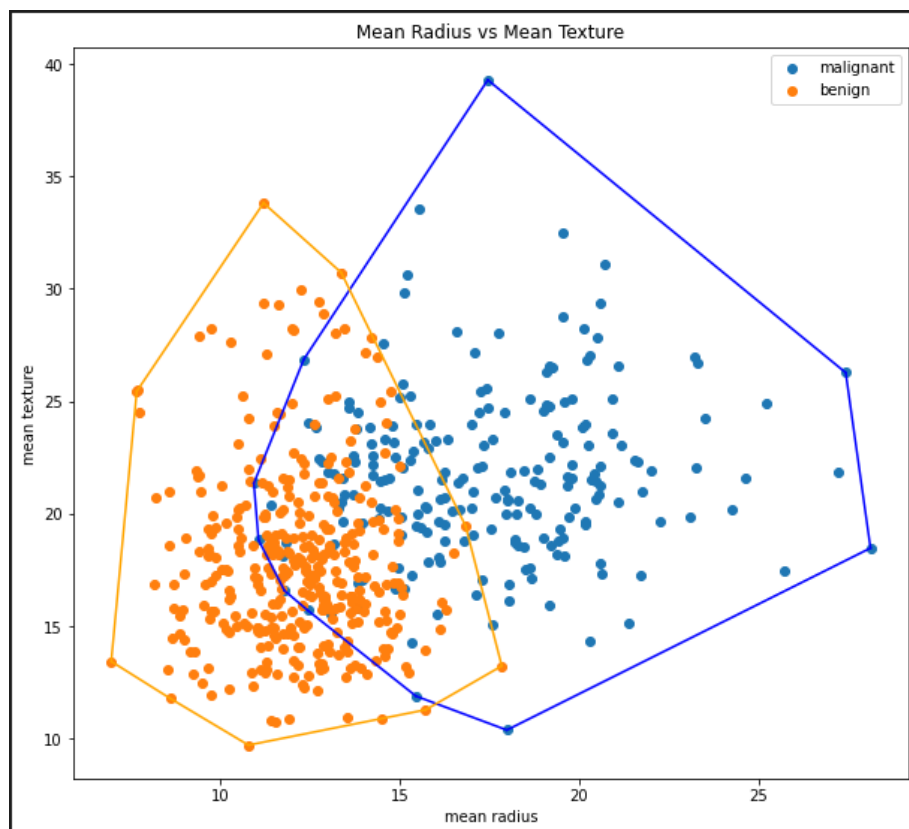




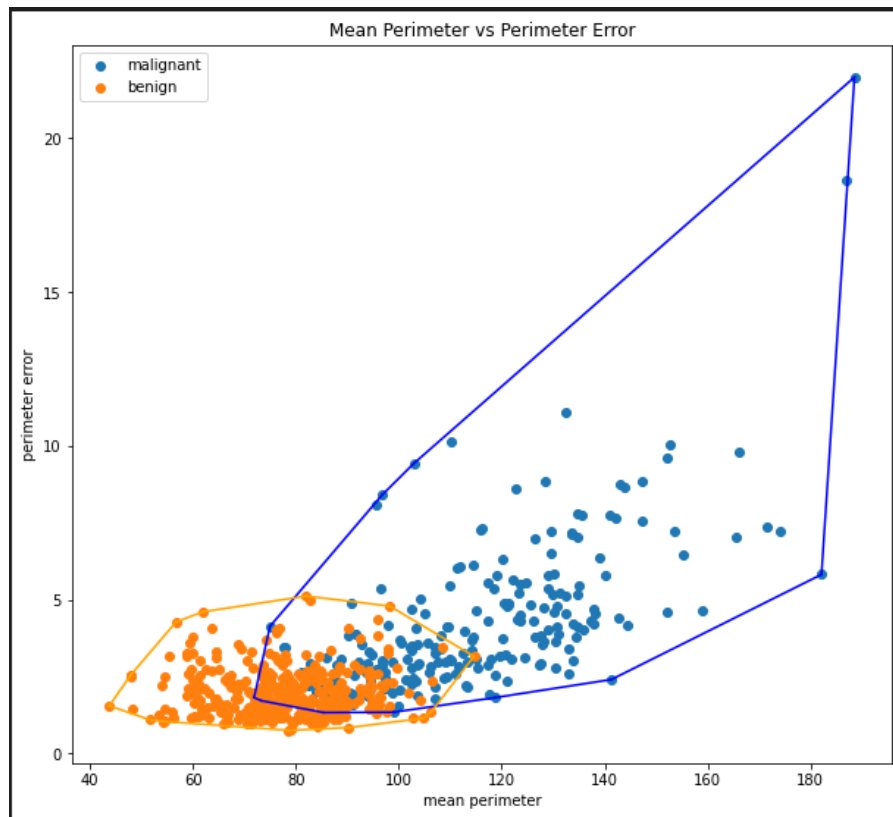
**Gambar 4.2.3.** Luaran program dengan parameter color\_intensity dan hue pada data wine

#### c. Data Breast Cancer

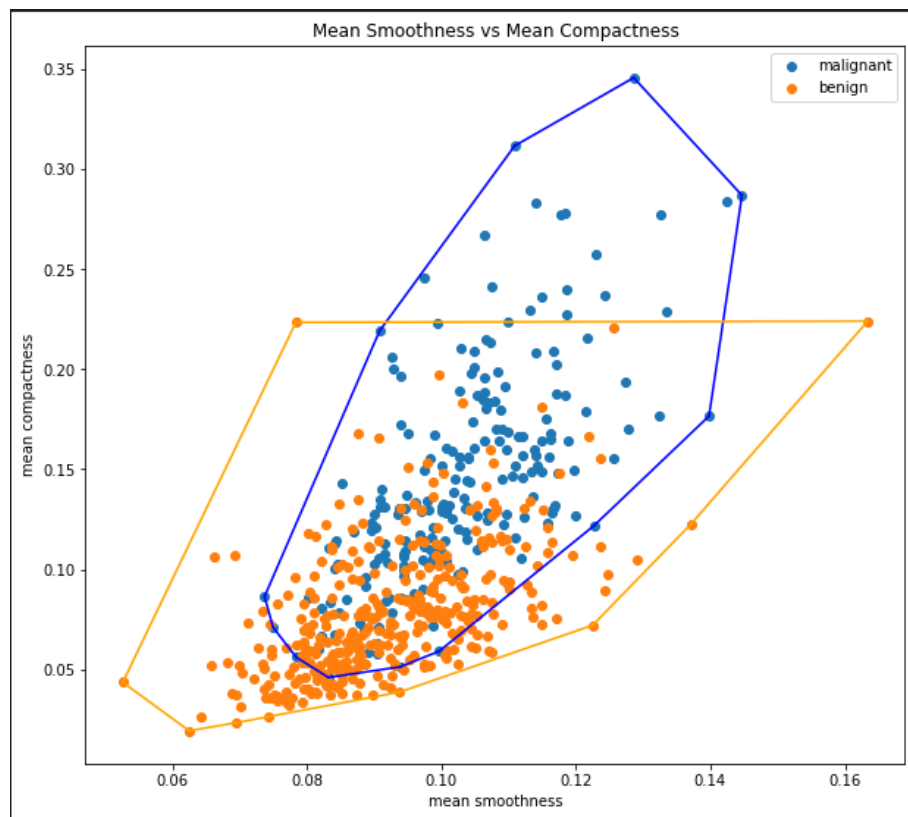
Berikut merupakan luaran program yang menggunakan data breast cancer



**Gambar 4.3.1.** Luaran program dengan parameter mean radius dan mean texture pada data breast cancer



**Gambar 4.3.2.** Luaran program dengan parameter mean perimeter dan perimeter error pada data breast cancer



**Gambar 4.3.2.** Luaran program dengan parameter mean smoothness dan mean compactness pada data breast cancer

## Bab 5 : Alamat Repository

Berikut merupakan alamat *repository* dari tugas ini.

[https://github.com/DeeGeeDow/Tucil2\\_13520152](https://github.com/DeeGeeDow/Tucil2_13520152)

**Tabel Keberjalanan Program**

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