

LAPORAN TUGAS KECIL

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

Ditujukan untuk memenuhi salah satu tugas kecil mata kuliah IF2211 Strategi Algoritma
pada Semester II Tahun Akademik 2021/2022

Disusun oleh:

Andhika Arta Aryanto (K3)

13520081



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

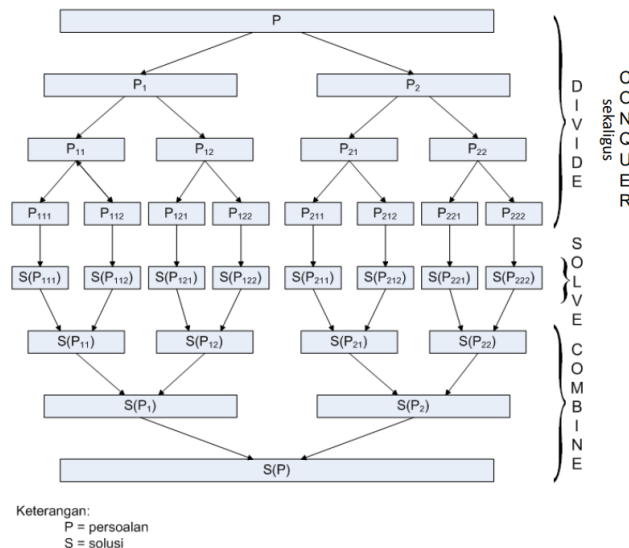
BAB I

PENJELASAN ALGORITMA PROGRAM

1.1 Divide and Conquer

Algoritma Divide and Conquer adalah strategi pemecahan masalah yang besar dengan cara melakukan pembagian masalah yang besar tersebut menjadi beberapa bagian yang lebih kecil secara rekursif hingga masalah tersebut dapat dipecahkan secara langsung, Solusi yang didapat dari setiap bagian kemudian digabungkan untuk membentuk sebuah solusi yang utuh. Untuk definisi Divide dan Conquer itu sendiri :

- Divide : membagi persoalan menjadi beberapa upa – persoalan yang memiliki kemiripan dengan persoalan semula namun berukuran lebih kecil (idealnya berukuran hampir sama)
- Conquer(solve) : menyelesaikan masing – masing upa-persoalan (secara langsung jika sudah berukuran kecil atau secara rekursif jika masih berukuran besar)
- Combine : menggabungkan solusi masing – masing upa – persoalan sehingga membentuk solusi persoalan semula



Obyek persoalan yang dibagi : masukan (input) atau *instances* persoalan yang berukuran n seperti : tabel, matriks, eksponen, polinom, dan masih banyak lagi. Tiap upa persoalan setelah dilakukan divide memiliki karakteristik yang sama dengan karakteristik persoalan semula sehingga metode *Divide and Conquer* lebih natural diungkapkan dalam skema rekursif sebagai berikut :

```
procedure DIVIDEandCONQUER(input P : problem, n : integer)
{ Menyelesaikan persoalan P dengan algoritma divide and conquer
  Masukan: masukan persoalan P berukuran n
  Luaran: solusi dari persoalan semula }
Deklarasi
  r : integer

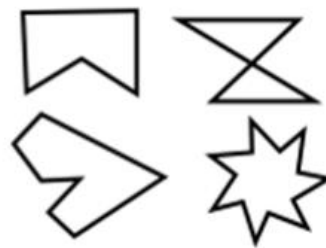
Algoritma
  if  $n \leq n_0$  then {ukuran persoalan P sudah cukup kecil}
    SOLVE persoalan P yang berukuran n ini
  else
    DIVIDE menjadi r upa-persoalan,  $P_1, P_2, \dots, P_r$ , yang masing-masing berukuran  $n_1, n_2, \dots, n_r$ 
    for masing-masing  $P_1, P_2, \dots, P_r$ , do
      DIVIDEandCONQUER( $P_i, n_i$ )
    endfor
    COMBINE solusi dari  $P_1, P_2, \dots, P_r$  menjadi solusi persoalan semula
  endif
```

1.2 Menyelesaikan persoalan Convex Hull dengan Divide and Conquer

Himpunan titik pada bidang planar disebut convex jika untuk sembarang dua titik pada bidang tersebut (misal p dan q), seluruh segmen garis yang berakhir di p dan q berada pada himpunan tersebut



Gambar 1: convex



Gambar 2: non convex

Pada tucil kali ini, diberikan tugas untuk membuat library yang dapat menyelesaikan persoalan Convex Hull menggunakan algoritma Divide and Conquer. Secara umum, algoritma ini akan mencari dua titik terjauh dan dilakukan divide dengan membagi titik – titik menjadi 2 yaitu bagian atas convex dan bawah, setelah dilakukan pembagian, Conquer dilakukan dengan pencarian titik terjauh untuk dimasukkan ke dalam convex, hal ini akan dilakukan terus menerus sampai tidak ada lagi titik yang masuk menjadi bagian atas/bawah convex. Library berhasil dibuat dengan fungsi utama bernama ConvexHull. Berikut langkah detail yang dilakukan dalam library yang telah dibuat :

1. Fungsi ConvexHull menerima sekumpulan titik (ndarray) yang berasal dari suatu dataset
2. Dibuat array *solution* untuk menampung titik – titik yang akan menjadi convex hull
3. Dilakukan quicksort pada array menggunakan library python berdasarkan absis yang menaik (ordinat akan dibandingkan apabila absis sama)
4. Setelah sort dilakukan, index pertama dan terakhir menjadi p_1 dan p_n , dimasukkan ke array solusi, dan dihapus dari kumpulan titik awal

5. Digunakan fungsi *aboveOrBelow* untuk membuat 2 array baru , satu untuk yang berada diatas garis $p1-pn$ dan satu lagi untuk yang dibawah
6. Pengelompokkan ini dilakukan dengan membuat garis $p1-pn$ menjadi fungsi garis $y = mx + c$ dan melakukan substitusi titik yang dikategorikan. Apabila hasil yang didapat adalah $y_0 > mx_0 + c$ berarti titik tersebut berada diatas garis dan dimasukkan ke array $s1$, dan sebaliknya akan dimasukkan ke array $s2$
7. Kedua array tersebut akan dimasukkan kedalam fungsi *recursiveconvex*, fungsi ini adalah fungsi rekursif yang akan terus menerus melakukan Divide and Conquer sampai list yang diterima sudah kosong (menandakan semua titik sudah diproses dan rekursif berhenti)
8. Pada awalnya akan dicari titik terjauh dari garis $p1-pn$, dan titik tersebut diberi nama *farthest* dan dimasukkan ke array solusi serta dihapus dari array awal
9. Setelah ditemukan , fungsi *aboveOrBelow* digunakan lagi, namun untuk garis yang berbeda yaitu $p1-farthest$ dan $farthest-p2$. Lalu fungsi *recursiveconvex* akan dipanggil lagi untuk garis dan array titik yang baru
10. Fungsi *recursiveconvex* ini akan terus dipanggil , dengan setiap iterasi akan menghasilkan *farthest* baru untuk dimasukkan ke dalam kumpulan solusi dan membuat *farthest* menjadi $p1$, $p2$ berikutnya.
11. Fungsi rekursif ini akan berakhir saat menerima array kosong, menandakan sudah tidak ada titik yang berada diatas ataupun bawah garis yang baru dibuat, menandakan garis sudah merupakan bagian paling luar dan merupakan convex untuk kumpulan titik tersebut
12. Kumpulan titik disort berdasarkan sudutnya dengan sebuah titik tengah agar membentuk array yang clockwise. Hal ini dilakukan untuk keperluan plotting. Setelah itu akan dikembalikan array solusi yang berisi semua titik yang merupakan bagian dari convex

Setelah array dari titik yang menjadi convex dibuat, terdapat file *main.py* yang memiliki tugas untuk menerima dataset, mengubah dataset menjadi kumpulan titik, memasukkan kumpulan titik ke fungsi yang telah dibuat, dan menampilkan hasil ConvexHull yang telah dicari.

Dalam library, terdapat fungsi – fungsi lain yang membantu pencarian convex, berikut fungsi – fungsi tersebut :

FUNGSI	KEGUNAAN
ConvexHull(numpyarray)	Fungsi utama yang akan mengembalikan kumpulan titik yang merupakan bagian dari convex hull
recursiveconvex(p1,p2,points,int)	Fungsi rekursif yang melakukan Divide and Conquer sampai semua titik berhasil diproses

aboveOrBelow(p1,p2,array)	Mencari kumpulan titik yang berada diatas atau dibawah garis p1-p2
distance(p1,p2,p3)	Menghitung jarak antara titik p3 dengan garis p1p2
farthestpoint(p1,p2,c)	Mencari titik terjauh pada array c dari garis p1p2
findAngle(p1,pmax,p2)	Mencari sudut yang dibentuk dari 3 titik p1,pmax,dan p2

BAB II

IMPLEMENTASI PROGRAM

2.1 Source Program dalam Python

2.1.1 Library ConvexHull.py

```
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import math
5  from sklearn import datasets
6
7  def ConvexHull(numpyarray) :
8      solution = []
9      points = numpyarray.tolist()
10
11      #Sort array untuk mencari 2 titik terjauh awal
12      sortedArray = sorted(points , key=lambda k: [k[0], k[1]])
13      p1 = sortedArray[0]
14      pn = sortedArray[-1]
15      solution += [p1,pn]
16
17      #Setelah dimasukkan ke array solusi, hilangkan kedua titik
18      sortedArray.pop(0)
19      sortedArray.pop(-1)
20
21      #Membagi kumpulan titik menjadi convex hull atas dan bawah
22      s1,s2 = aboveOrBelow(p1,pn,sortedArray)
23
24      #Dilakukan rekursi(divide terus menerus)
25      solution += recursiveconvex(p1,pn,s1,1)
26      solution += recursiveconvex(p1,pn,s2,2)
27
28      solution = np.asarray(solution)
29
30      #Agar garis yang dibentuk membentuk poligon mengelilingi sempurna,
31      #titik - titik di sort berdasarkan sudut yang dibuat dengan titik t
32      #engah
33      x = [p[0] for p in solution]
34      y = [p[1] for p in solution]
35      center = (sum(x) / len(solution), sum(y) / len(solution))
36      solution = sorted(solution,key = lambda k: (math.atan2(k[1]-
37      center[1],k[0]-center[0])))
38      solution.append(solution[0])
39      return solution
```

```
1 def recursiveconvex(p1,p2,points,int) :
2
3     #Fungsi rekursif untuk mencari titik2 convex hull lainnya, int disini
4     #menandakan arah , 1 berarti atas(kasus convex hull atas) , 2 bawah
5     # (convex hull bawah)
6     #Exit case untuk rekursif
7     if points == [] or p1 is None or p2 is None :
8         return []
9
10    solution = []
11    #Mencari titik terjauh dari garis
12    farthest = farthestpoint(p1,p2,points)
13    solution += [farthest]
14    points.remove(farthest)
15
16    #Membagi kumpulan titik menjadi atas dan bawah dengan garis yang baru
17    #dibuat
18    above1, below1 = aboveOrBelow(p1,farthest,points)
19    above2, below2 = aboveOrBelow(p2,farthest,points)
20
21    #Rekursif
22    if int == 1 :
23        solution += recursiveconvex(p1,farthest,above1,1)
24        solution += recursiveconvex(farthest,p2,above2,1)
25    else :
26        solution += recursiveconvex(p1,farthest,below1,2)
27        solution += recursiveconvex(farthest,p2,below2,2)
28
29    return solution
```

```
1 def distance(p1,p2,p3) :
2     #fungsi menghitung jarak dari garis p1-p2 ke titik p3
3     p1 = np.asarray(p1)
4     p2 = np.asarray(p2)
5     p3 = np.asarray(p3)
6     return np.abs(np.cross(p2-p1, p1-p3)) / np.linalg.norm(p2-p1)
7
8 def farthestpoint(p1,p2,c) :
9     #Mencari titik terjauh ke garis p1-p2 dari kumpulan titik c
10    max = -1
11    farthest = None
12    for point in c :
13        temp = distance(p1,p2,point)
14        if temp > max :
15            max = temp
16            farthest = point
17        elif temp == max :
18            if(findAngle(p1,point,p2) > findAngle
19               (p1,farthest,p2)) :
20                farthest = point
21                max = temp
22    return farthest
23
24 def findAngle(p1,pmax,p2) :
25
26    #mencari sudut antara ketiga titik, untuk kasus ini berarti sudut
27    antara garis p1-pmax dan pmax-p2
28    ang = math.degrees(math.atan2(p2[1]-pmax[1], p2[0]-pmax[0]) -
29                          math.atan2(p1[1]-pmax[1], p1[0]-pmax[0]))
30    return ang + 360 if ang < 0 else ang
```

```
1 def aboveOrBelow(p1,p2,array) :
2
3     #Fungsi untuk mencari kumpulan titik yang menjadi bagian atas atau b
4     awah suatu garis dalam convex hull
5     s1 = []
6     s2 = []
7
8     #X sama, garis vertikal tidak ada atas bawah
9     if p2[0] - p1[0] == 0 :
10        return s1,s2
11
12    #Membuat garis menjadi rumus umum y = mx + c, setiap poin akan disub
13    situsi, apabila y0 > mx0 + c berarti dia diatas dan sebaliknya
14    grad = (p2[1] - p1[1]) / (p2[0] - p1[0])
15    c = p1[1] - (grad * p1[0])
16    for points in array :
17        if points[1] < grad * (points[0]) + c :
18            s2.append(points)
19        if points[1] > grad * (points[0]) + c :
20            s1.append(points)
21    return s1,s2
```


2.2.1 main.py

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from sklearn import datasets
5 import matplotlib.pyplot as plt
6 from ConvexHull import ConvexHull
7 from os import path
8
9 print("=====")
10 print("Pilihan dataset :")
11 print("DATASET SCIKIT LEARN")
12 print("1 Iris")
13 print("2 Wine")
14 print("3 Breast Cancer")
15 print(
16     "Program ini juga bisa menerima file CSV, untuk melakukan ini mas
17     kkan CSV ke folder test, pastikan semua value dalam csv merupakan
18     nomor"
19 )
20
21 cc = int(input())
22
23 if (cc == 1) :
24     data = datasets.load_iris()
25     df = pd.DataFrame(data.data, columns=data.feature_names)
26     df['Target'] = pd.DataFrame(data.target)
27     hastarget = True
```

```
28 elif (cc == 2) :
29     data = datasets.load_wine()
30     df = pd.DataFrame(data.data, columns=data.feature_names)
31     df['Target'] = pd.DataFrame(data.target)
32     hastarget = True
33 elif (cc == 3) :
34     data = datasets.load_breast_cancer()
35     df = pd.DataFrame(data.data, columns=data.feature_names)
36     df['Target'] = pd.DataFrame(data.target)
37     hastarget = True
38 elif (cc == 4) :
39     data = input("Nama file csv = ")
40     file = ".\\test\\" + data
41     while not path.exists(file) :
42         print("File tidak dapat ditemukan")
43         data = input("Nama file csv = ")
44         file = ".\\test\\" + data
45     df = pd.read_csv(file)
46     if ('target' in df or 'Target' in df or 'TARGET' in df) :
47         hastarget = True
48     if (hastarget == False) :
49         print(
50             "CSV TIDAK ADA TARGET ! pilih 5 untuk csv tanpa target"
51         )
52 elif (cc == 5) :
53     hastarget = False
54     data = input("Nama file csv = ")
55     file = ".\\test\\" + data
56     while not path.exists(file) :
57         print("File tidak dapat ditemukan")
58         data = input("Nama file csv = ")
59         file = ".\\test\\" + data
60     df = pd.read_csv(file)
61     df.head()
```

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

```
1  if(hastarget) :
2      if(cc == 4) :
3          print(
4              "=====")
5          print("Kolom - kolom pada dataset :")
6          col = df.columns.values.tolist()
7          i = 1
8          for col_names in col :
9              if (str.lower(col_names) != "target") :
10                 print(f"{i}.{col_names}")
11                 i += 1
12
13             print(
14                 "=====")
15             print("Pilih 2 kolom Yang ingin di plot (dalam angka):")
16             X = 0
17             Y = 0
18             while (X <= 0 or X >= len(col)) :
19                 X = int(input("Kolom Yang menjadi X : "))
20             while (Y <= 0 or Y >= len(col)) :
21                 Y = int(input("Kolom Yang menjadi Y : "))
22
23             ## 4. INISIALISASI PYPLLOT
24             plt.figure(figsize = (10, 6))
25             plt.title(label='Convex Hull', fontsize=20)
26             xname = col[X-1]
27             yname = col[Y-1]
28             plt.title(f'{xname} vs {yname}')
29             plt.xlabel(col[X-1])
30             plt.ylabel(col[Y-1])
31             target = df.target.unique()
32             print(X)
33             print(Y)
34
35             ## 5. MEMBUAT CONVEXHULL
36             for i in range(len(target)) :
37                 bucket = df[df['target'] == target[i]]
38                 bucket = bucket.iloc[:, [X-1, Y-1]].values
39                 hull = ConvexHull(bucket)
40                 plt.scatter(bucket[:, 0], bucket[:, 1], label=
41                     target[i])
42
43             Xarray, Yarray = zip(*hull)
44             plt.plot(Xarray, Yarray, '-o')
45
46             elif(cc == 1 or cc == 2 or cc == 3) :
```

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

```
40     print(
41         "=====")
42     print("Kolom - kolom pada dataset :")
43     col = df.columns.values.tolist()
44     i = 1
45     for col_names in col :
46         if (str.lower(col_names) != "target") :
47             print(f"{i}.{col_names}")
48             i += 1
49
50     print(
51         "=====")
52     print("Pilih 2 kolom Yang ingin di plot (dalam angka):")
53     X = 0
54     Y = 0
55     while (X <= 0 or X >= len(col)) :
56         X = int(input("Kolom Yang menjadi X : "))
57     while (Y <= 0 or Y >= len(col)) :
58         Y = int(input("Kolom Yang menjadi Y : "))
59     plt.figure(figsize = (10, 6))
60     xname = data.feature_names[X-1]
61     yname = data.feature_names[Y-1]
62     plt.title(f'{xname} vs {yname}')
63     plt.xlabel(xname)
64     plt.ylabel(yname)
65     for i in range(len(data.target_names)):
66         bucket = df[df['Target'] == i]
67         bucket = bucket.iloc[:,[X-1,Y-1]].values
68         hull = ConvexHull(bucket)
69
70     #bagian ini diganti dengan hasil implementasi
71     #ConvexHull Divide & Conquer
72     plt.scatter(bucket[:, 0], bucket[:, 1], label=
73     data.target_names[i])
74     print(hull)
75     x, y = zip(*hull)
76
77     plt.plot(x, y, '-o')
78     plt.legend()
79     plt.show()
```

```
1 elif(hastarget == False and cc == 5) :
2     print(
3         "=====")
4     print("Kolom - kolom pada dataset :")
5     col = df.columns.values.tolist()
6     i = 1
7     for col_names in col :
8         if (str.lower(col_names) != "target") :
9             print(f"{i}.{col_names}")
10             i += 1
11
12     print(
13         "=====")
14     print("Pilih 2 kolom Yang ingin di plot (dalam angka):")
15     X = 0
16     Y = 0
17     while (X <= 0 or X >= len(col)) :
18         X = int(input("Kolom Yang menjadi X : "))
19     while (Y <= 0 or Y >= len(col)) :
20         Y = int(input("Kolom Yang menjadi Y : "))
21
22     ## 4. INISIALISASI PYPLOT
23     plt.figure(figsize = (10, 6))
24     plt.title(label='ConveX Hull', fontsize=20)
25     plt.xlabel(col[X-1])
26     plt.ylabel(col[Y-1])
27     print(X)
28     print(Y)
29
30     ## 5. MEMBUAT CONVEXHULL
31
32     bucket = df.iloc[:,[X-1,Y-1]].values
33     hull = ConvexHull(bucket)
34     plt.scatter(bucket[:, 0], bucket[:, 1])
35     Xarray, Yarray = zip(*hull)
36     plt.plot(Xarray, Yarray, '-o')
37     plt.show()
```

BAB III

PENGUJIAN PROGRAM

1. Input dan Output Pengujian Program

Dilakukan pengujian library convex hull untuk 6 dataset berbeda, 3 dataset berasal dari modul python scikit learn dan 3 dataset lagi merupakan 3 CSV. Akan diambil 2 pasangan atribut random yang akan diplot dan dicari convex hullnya. Untuk dataset iris diambil atribut yang diminta oleh spek tucil ini , yaitu petal length vs petal width dan sepal length vs sepal width. Berikut hasil pengujian :

3.1.1 Interaksi awal Program

Berikut interaksi yang akan dilakukan pengguna saat awal memakai program

```
=====
Pilihan dataset :
DATASET SCIKIT LEARN
1 Iris
2 Wine
3 Breast Cancer
Program ini juga bisa menerima file CSV, untuk melakukan ini masukkan CSV ke folder test, pastikan semua value dalam csv merupakan nomor
4. CSV sendiri (ada target)
5. CSV sendiri (tidak ada target)
=====
1
=====
Kolom - kolom pada dataset :
1.sepal length (cm)
2.sepal width (cm)
3.petal length (cm)
4.petal width (cm)
=====
Pilih 2 kolom Yang ingin di plot (dalam angka):
Kolom Yang menjadi X : 1
Kolom Yang menjadi Y : 2
```

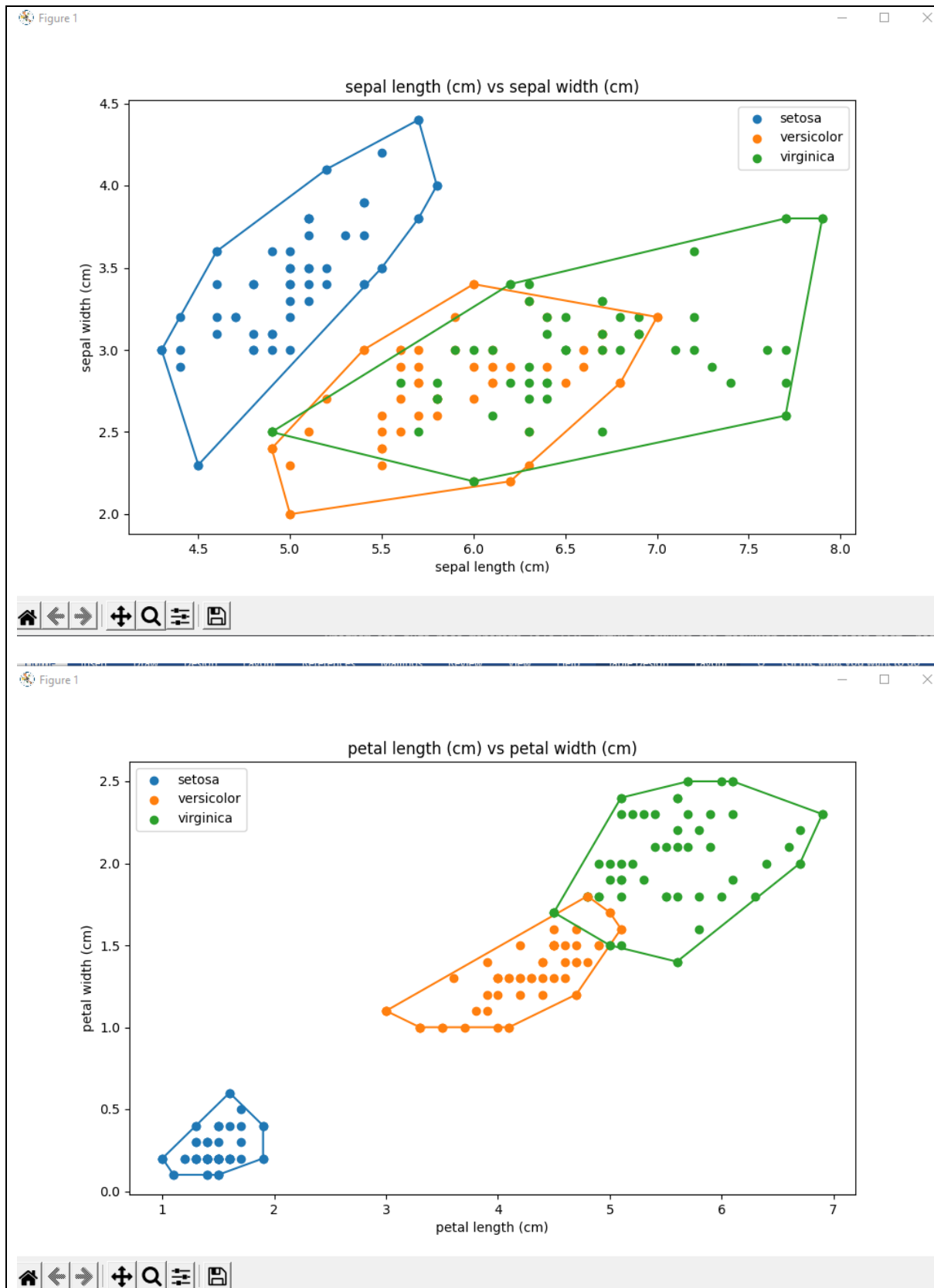
```
=====
Pilihan dataset :
DATASET SCIKIT LEARN
1 Iris
2 Wine
3 Breast Cancer
Program ini juga bisa menerima file CSV, untuk melakukan ini masukkan CSV ke folder test, pastikan semua value dalam csv merupakan nomor
4. CSV sendiri (ada target)
5. CSV sendiri (tidak ada target)
=====
4
Nama file csv = heart.csv
=====
Kolom - kolom pada dataset :
1.age
2.sex
3.cp
4.trestbps
5.chol
6.fbs
7.restecg
8.thalach
9.exang
10.oldpeak
11.slope
12.ca
13.thal
=====
Pilih 2 kolom Yang ingin di plot (dalam angka):
Kolom Yang menjadi X : 3
Kolom Yang menjadi Y : 4
3
4
```

3.1.2 Pengujian Dataset

<i>Nama Dataset :</i> Iris	<i>Pasangan Atribut :</i> sepal-length, sepal-width petal-length, petal-width
<i>Input</i>	

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	\
0	5.1	3.5	1.4	0.2	
1	4.9	3.0	1.4	0.2	
2	4.7	3.2	1.3	0.2	
3	4.6	3.1	1.5	0.2	
4	5.0	3.6	1.4	0.2	
..	
145	6.7	3.0	5.2	2.3	
146	6.3	2.5	5.0	1.9	
147	6.5	3.0	5.2	2.0	
148	6.2	3.4	5.4	2.3	
149	5.9	3.0	5.1	1.8	
	Target				
0	0				
1	0				
2	0				
3	0				
4	0				
..	...				
145	2				
146	2				
147	2				
148	2				
149	2				
[150 rows x 5 columns]					

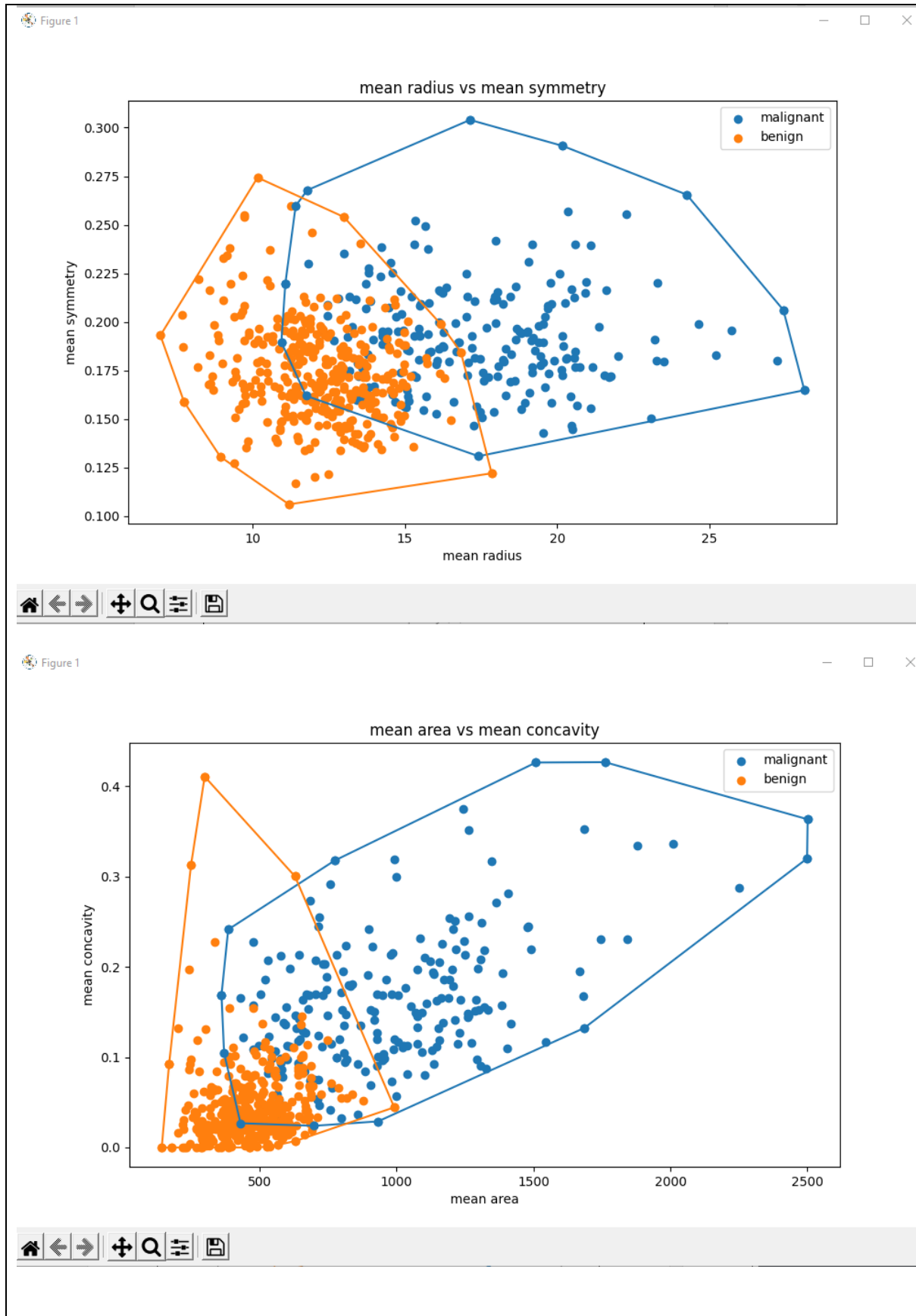
Output



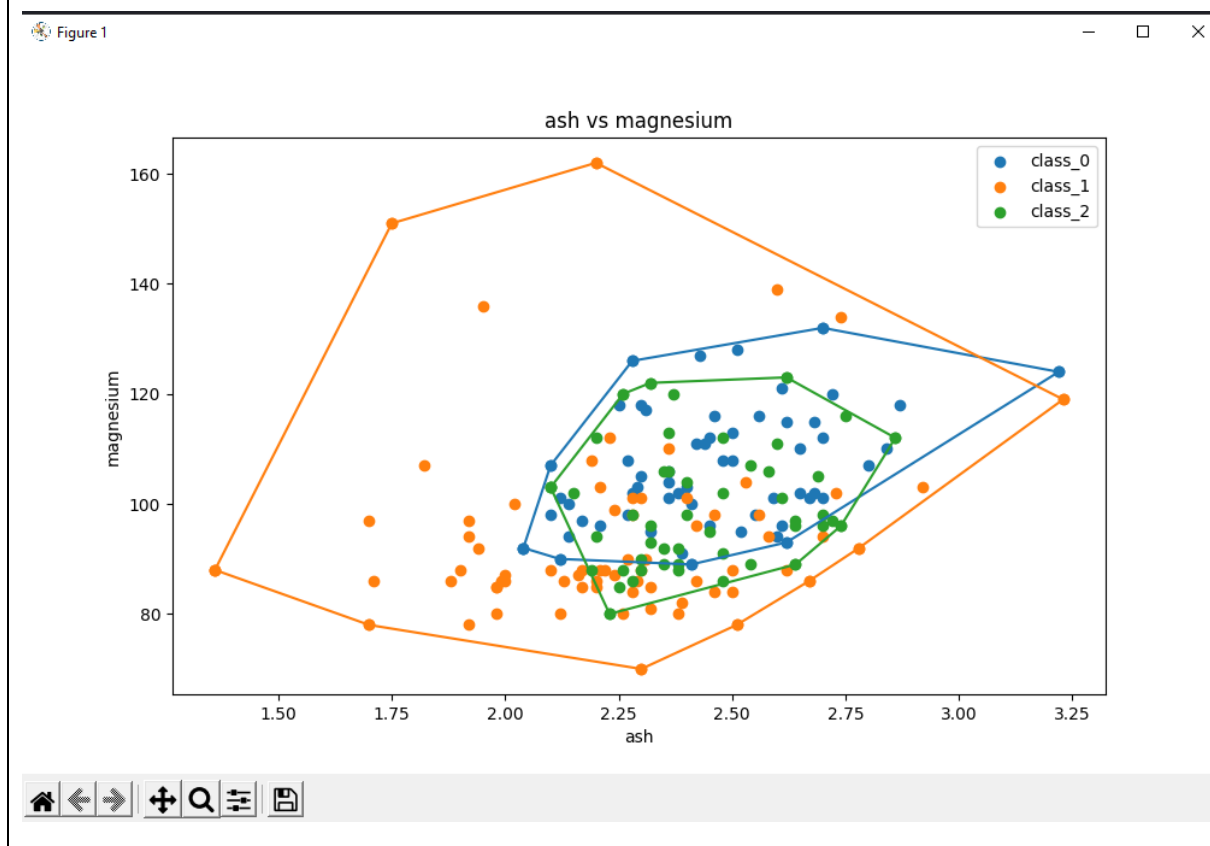
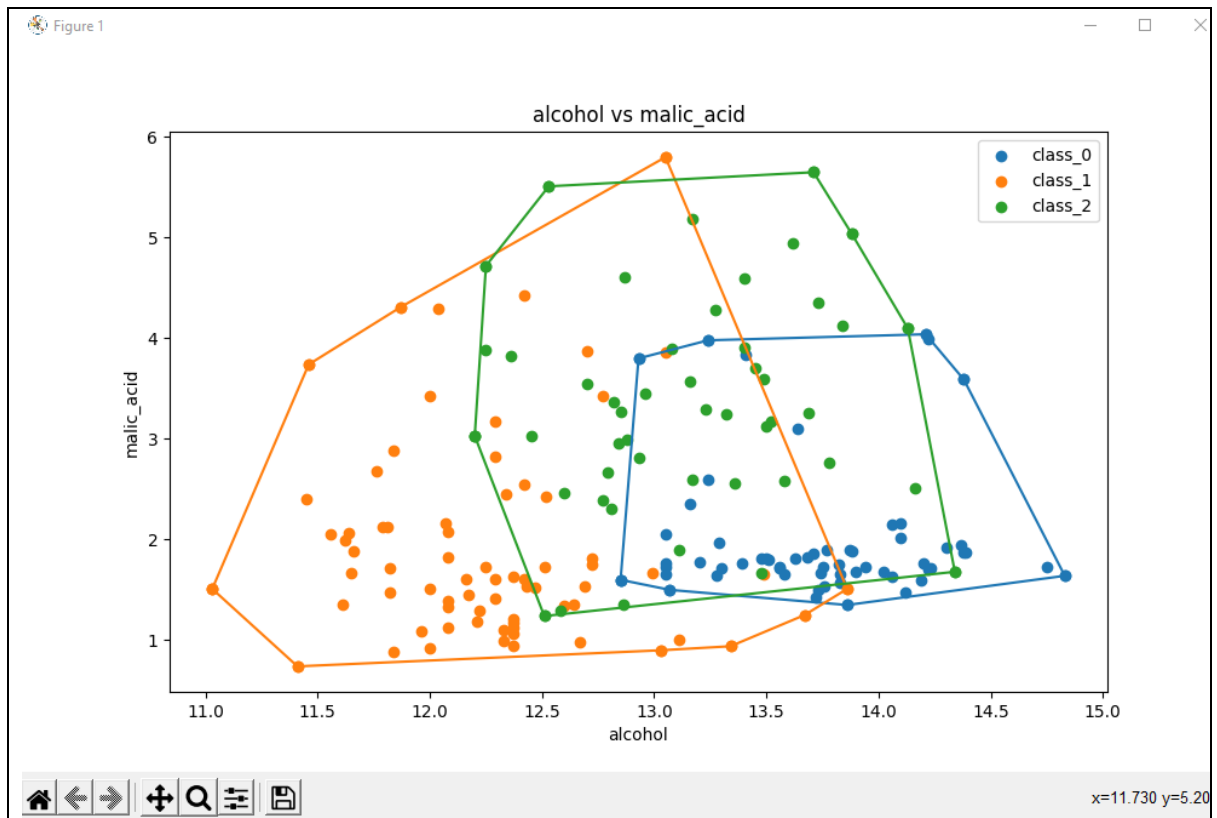
Nama Dataset :
Breast Cancer

Pasangan Atribut :
mean-radius, mean-symmetry

				mean-area, mean concavity	
Input					
	mean radius	mean texture	mean perimeter	mean area	mean smoothness \
0	17.99	10.38	122.80	1001.0	0.11840
1	20.57	17.77	132.90	1326.0	0.08474
2	19.69	21.25	130.00	1203.0	0.10960
3	11.42	20.38	77.58	386.1	0.14250
4	20.29	14.34	135.10	1297.0	0.10030
..
564	21.56	22.39	142.00	1479.0	0.11100
565	20.13	28.25	131.20	1261.0	0.09780
566	16.60	28.08	108.30	858.1	0.08455
567	20.60	29.33	140.10	1265.0	0.11780
568	7.76	24.54	47.92	181.0	0.05263
	mean compactness	mean concavity	mean concave points	mean symmetry \	
0	0.27760	0.30010	0.14710	0.2419	
1	0.07864	0.08690	0.07017	0.1812	
2	0.15990	0.19740	0.12790	0.2069	
3	0.28390	0.24140	0.10520	0.2597	
4	0.13280	0.19800	0.10430	0.1809	
..	
564	0.11590	0.24390	0.13890	0.1726	
565	0.10340	0.14400	0.09791	0.1752	
566	0.10230	0.09251	0.05302	0.1590	
567	0.27700	0.35140	0.15200	0.2397	
568	0.04362	0.00000	0.00000	0.1587	
...					
566	0.1418	0.2218		0.07820	0
567	0.2650	0.4087		0.12400	0
568	0.0000	0.2871		0.07039	1
[569 rows x 31 columns]					
Output					

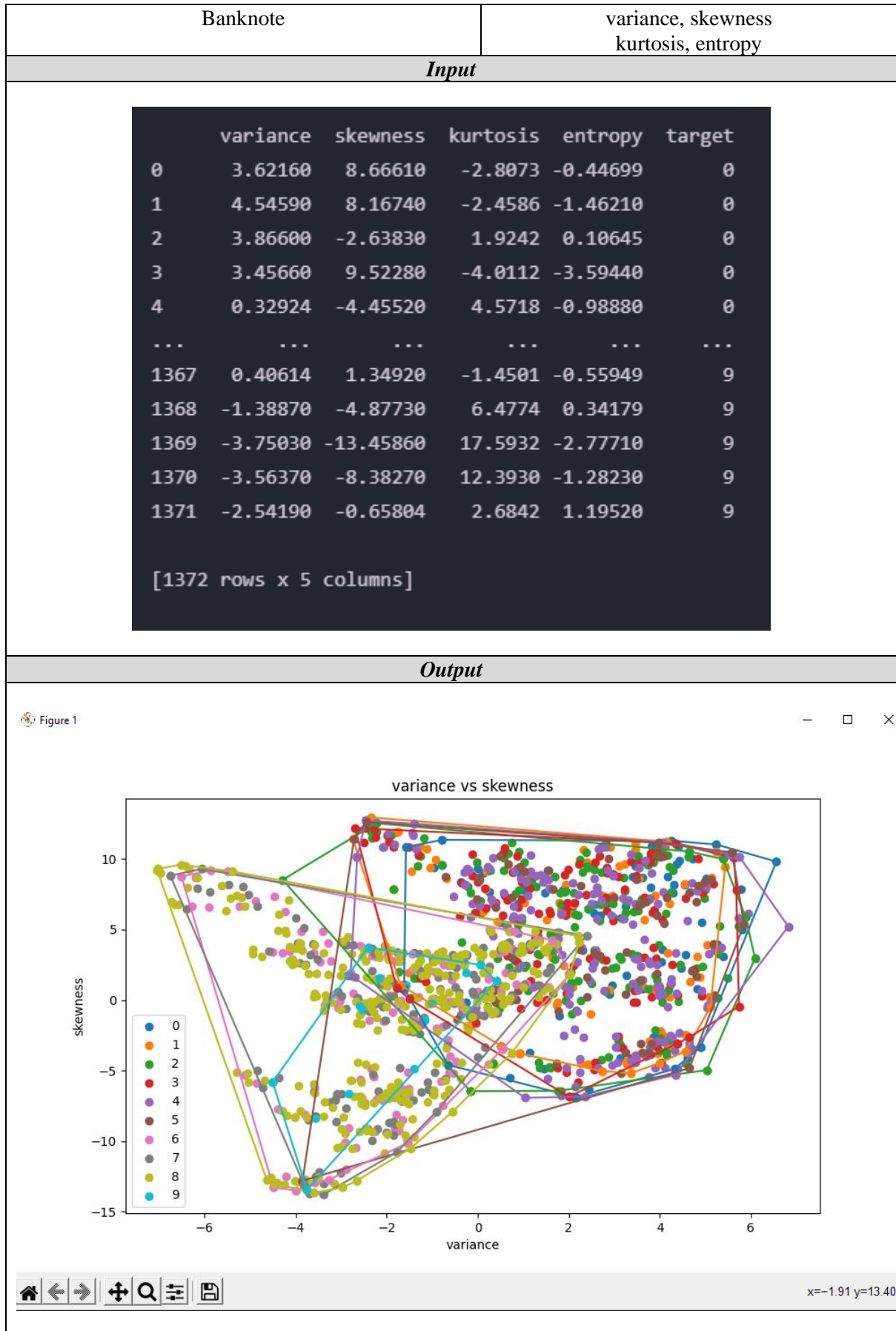


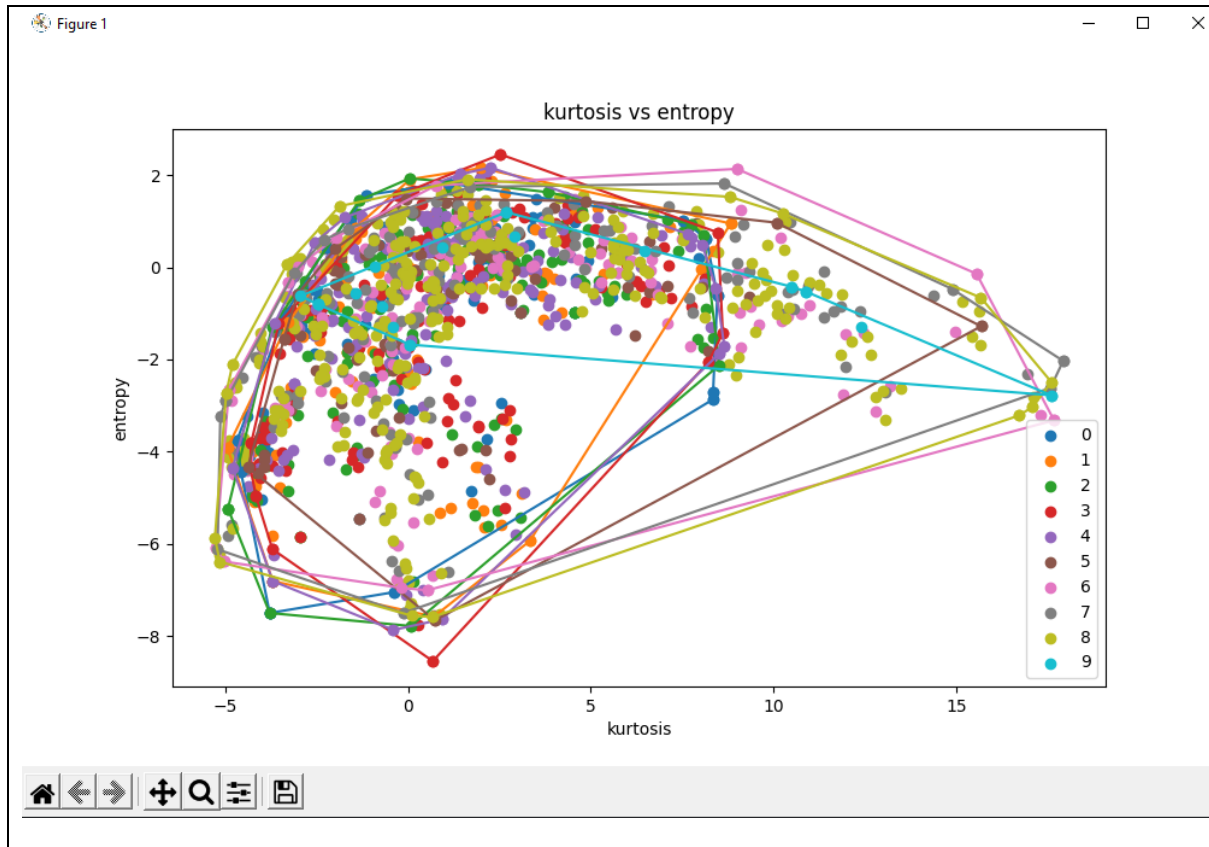
Nama Dataset : Wine				Pasangan Atribut : alcohol, malic-acid ash,magnesium			
Input							
	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	\
0	14.23	1.71	2.43	15.6	127.0	2.80	
1	13.20	1.78	2.14	11.2	100.0	2.65	
2	13.16	2.36	2.67	18.6	101.0	2.80	
3	14.37	1.95	2.50	16.8	113.0	3.85	
4	13.24	2.59	2.87	21.0	118.0	2.80	
..	
173	13.71	5.65	2.45	20.5	95.0	1.68	
174	13.40	3.91	2.48	23.0	102.0	1.80	
175	13.27	4.28	2.26	20.0	120.0	1.59	
176	13.17	2.59	2.37	20.0	120.0	1.65	
177	14.13	4.10	2.74	24.5	96.0	2.05	
	flavanoids	nonflavanoid_phenols		proanthocyanins	color_intensity	hue	\
0	3.06		0.28	2.29	5.64	1.04	
1	2.76		0.26	1.28	4.38	1.05	
2	3.24		0.30	2.81	5.68	1.03	
3	3.49		0.24	2.18	7.80	0.86	
4	2.69		0.39	1.82	4.32	1.04	
..	
173	0.61		0.52	1.06	7.70	0.64	
174	0.75		0.43	1.41	7.30	0.70	
175	0.69		0.43	1.35	10.20	0.59	
176	0.68		0.53	1.46	9.30	0.60	
177	0.76		0.56	1.35	9.20	0.61	
...							
175			1.56	835.0	2		
176			1.62	840.0	2		
177			1.60	560.0	2		
[178 rows x 14 columns]							
Output							



Nama Dataset :

Pasangan Atribut :





<i>Nama Dataset :</i> Heart	<i>Pasangan Atribut :</i> age, chol trestbps, exang
<i>Input</i>	

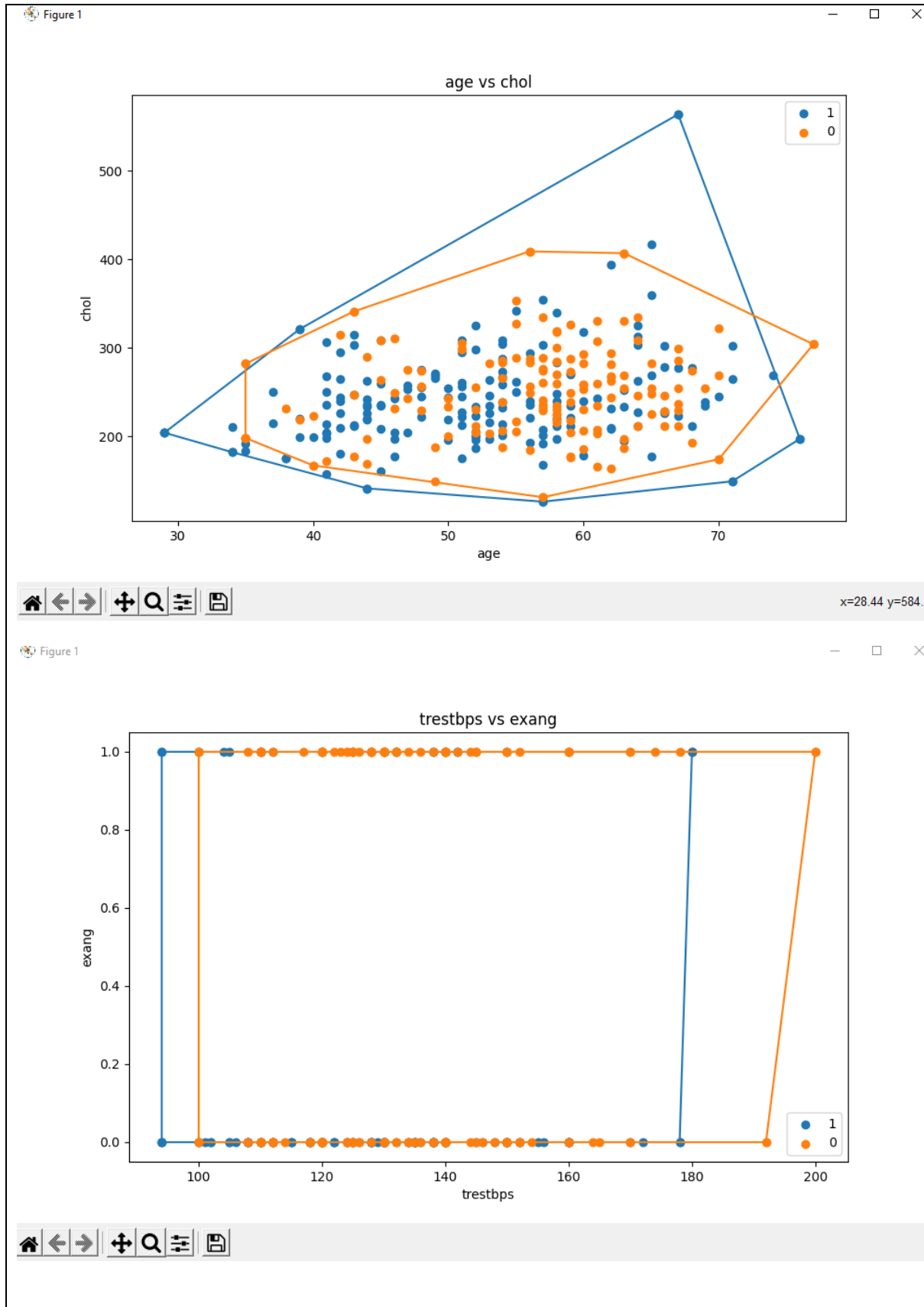
```

    age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     63   1   3      145   233   1         0    150     0      2.3
1     37   1   2      130   250   0         1    187     0      3.5
2     41   0   1      130   204   0         0    172     0      1.4
3     56   1   1      120   236   0         1    178     0      0.8
4     57   0   0      120   354   0         1    163     1      0.6
..     ...   ...  ..      ...   ...   ...       ...    ...     ...     ...
298    57   0   0      140   241   0         1    123     1      0.2
299    45   1   3      110   264   0         1    132     0      1.2
300    68   1   0      144   193   1         1    141     0      3.4
301    57   1   0      130   131   0         1    115     1      1.2
302    57   0   1      130   236   0         0    174     0      0.0

    slope  ca  thal  target
0         0  0    1        1
1         0  0    2        1
2         2  0    2        1
3         2  0    2        1
4         2  0    2        1
..     ...  ..   ...     ...
298      1  0    3        0
299      1  0    3        0
300      1  2    3        0
301      1  1    3        0
302      1  1    2        0

[303 rows x 14 columns]
```

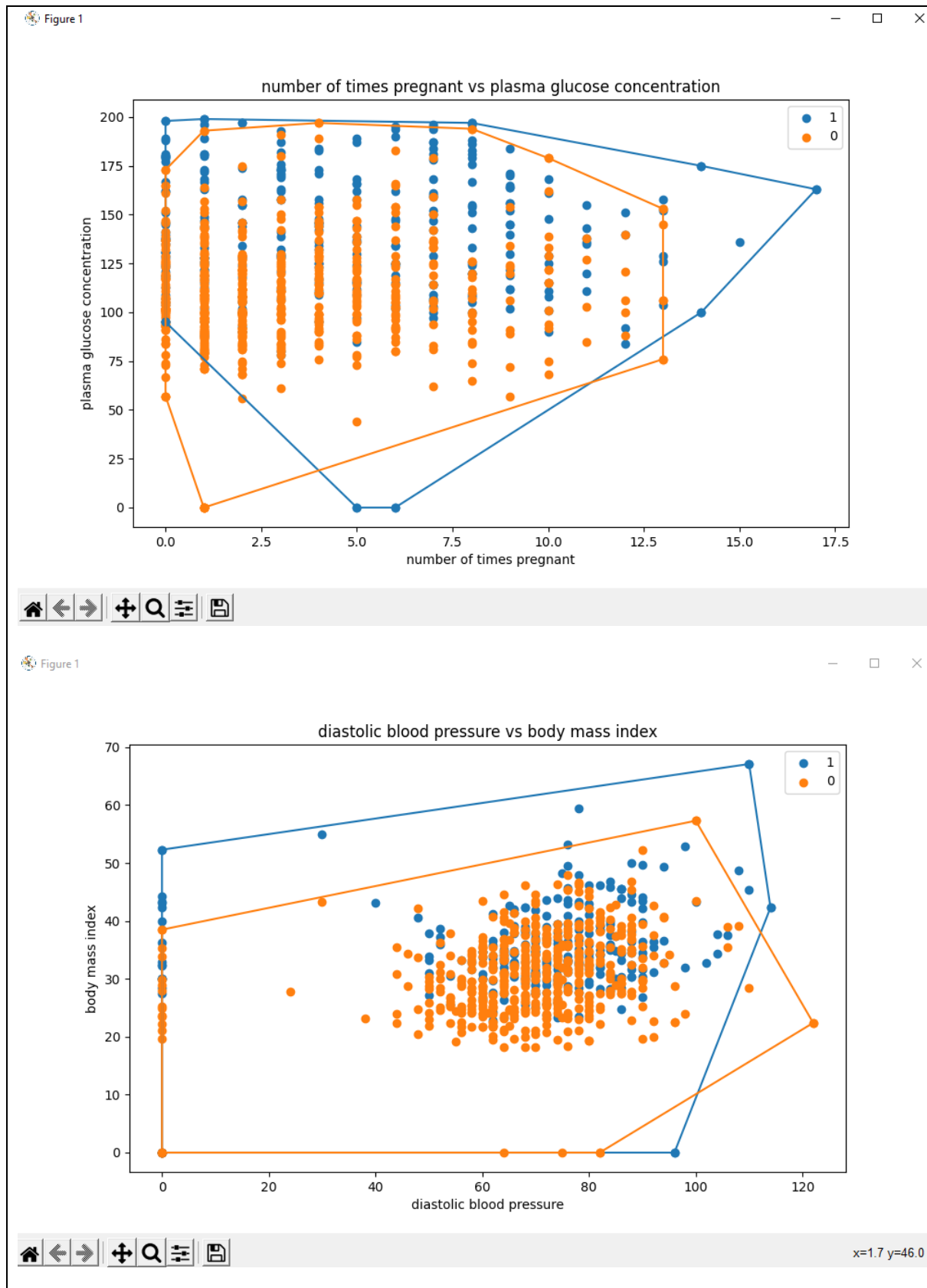
Output



Nama Dataset :

Pasangan Atribut :

Indiansdiabetes	number-of-times-pregnant, plasma-glucose-concentration diastolic-blood-pressure, body-mass-index
<i>Input</i>	
<pre> number of times pregnant plasma glucose concentration \ 0 6 148 1 1 85 2 8 183 3 1 89 4 0 137 763 10 101 764 2 122 765 5 121 766 1 126 767 1 93 diastolic blood pressure triceps skinfold thickness \ 0 72 35 1 66 29 2 64 0 3 66 23 4 40 35 763 76 48 764 70 27 765 72 23 766 60 0 767 70 31 ... 765 0 766 1 767 0 [768 rows x 9 columns]</pre>	
<i>Output</i>	



LAMPIRAN

1. REPOSITORY GITHUB :

<https://github.com/dhikaarta/Convex-Hull-Library-and-Visualizer>

2. CHECKLIST:

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

DAFTAR PUSTAKA

- Informatika.stei.itb.ac.id/~rinaldi.munir. (2022). Algoritma Divide and Conquer Bagian 1. Diakses pada 26 Februari 2022, dari [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-\(2021\)-Bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian1.pdf)
- Informatika.stei.itb.ac.id/~rinaldi.munir. (2022). Algoritma Divide and Conquer Bagian 4. Diakses pada 26 Februari 2022, dari [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Divide-and-Conquer-\(2022\)-Bagian4.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Divide-and-Conquer-(2022)-Bagian4.pdf)
- <https://www.geeksforgeeks.org/get-column-names-from-csv-using-python/>
- <https://www.analyticsvidhya.com/blog/2021/06/complete-guide-to-working-with-csv-files-in-python-with-pandas/>