

Tugas Kecil 2 IF2211 Strategi Algoritma

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



Disusun oleh:

13520016 - Gagas Praharsa Bahar

INSTITUT TEKNOLOGI BANDUNG

2022

DAFTAR ISI

DAFTAR ISI	2
BAB 1	3
Algoritma Divide and Conquer	3
Convex Hull	3
Algoritma Penyelesaian Convex Hull dengan Pendekatan Divide and Conquer	4
BAB 2	6
myConvexHull.py	6
main.py dan main.ipynb	6
Dependencies	7
BAB 3	7
Repository Program	7
Source Code Program	7
BAB 4	10
Interface Program	10
Dataset Iris	10
Dataset Wine	12
Dataset Breast Cancer	14
Dataset heart.csv	15
REFERENSI	17

BAB 1

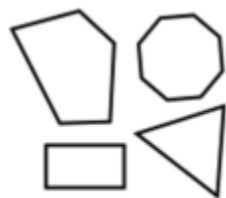
DESKRIPSI MASALAH DAN ALGORITMA

1.1 Algoritma Divide and Conquer

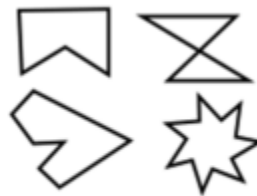
Algoritma *divide and conquer* merupakan sebuah pendekatan dalam pemecahan masalah. Pada algoritma divide and conquer, terdapat dua elemen utama, yaitu Divide dan Conquer. Untuk menyelesaikan permasalahan dengan menggunakan divide and conquer, pertama-tama permasalahan dapat dibagi menjadi beberapa upa-persoalan yang memiliki kemiripan persoalan semula namun berukuran lebih kecil terlebih dahulu. Idealnya, masing-masing upa-persoalan memiliki ukuran yang sama. Setelah membagi-bagi permasalahan menjadi bagian-bagian kecil, selesaikan masing-masing permasalahan (conquer) secara langsung atau secara rekursif apabila persoalan masih berukuran besar. Terakhir, gabungkan solusi masing-masing upa-persoalan sehingga membentuk solusi persoalan semula. Dikarenakan setiap upa-persoalan memiliki karakteristik yang sama, algoritma Divide and Conquer seringkali lebih natural diungkapkan dalam skema rekursif dan diimplementasikan dengan fungsi rekursif pula.

1.2 Convex Hull

Convex Hull adalah sebuah istilah dalam geometri. Himpunan titik dalam suatu bidang planar dikatakan convex apabila untuk sembarang dua titik pada bidang tersebut (misal p dan q), seluruh segmen garis yang berakhir di p dan q berada pada himpunan tersebut.

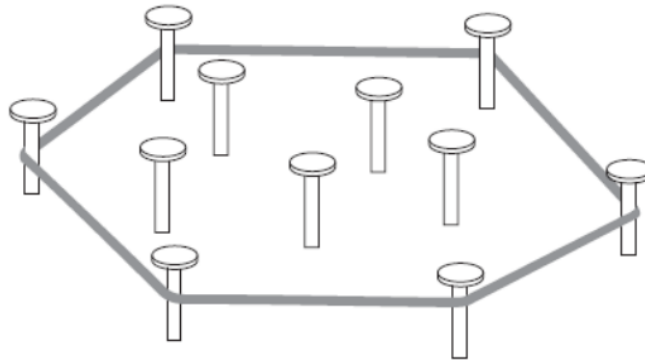


Gambar 1: convex



Gambar 2: non convex

Convex Hull dari himpunan titik S adalah himpunan convex terkecil (convex polygon) yang mengandung S . Convex Hull dapat dianalogikan seperti karet gelang yang mengelilingi beberapa titik pada suatu bidang, seperti contoh berikut:



Penentuan convex hull dari kumpulan titik adalah salah satu hal penting yang menjadi sorotan pada komputasi geometri. Hal ini disebabkan karena pemanfaatan dari convex hull ini sangat banyak, mulai dari *collision detection*, persoalan optimasi, pengelompokan data, dan pendeteksian *outliers* pada kumpulan data.

1.3 Algoritma Penyelesaian Convex Hull dengan Pendekatan Divide and Conquer

Dalam menyelesaikan permasalahan *word search puzzle* secara algoritmik, penulis menggunakan pendekatan *brute force*. Adapun langkah-langkah penyelesaian permasalahan dalam algoritma dapat dijelaskan secara deskriptif sebagai berikut:

1. Urutkan titik-titik berdasarkan posisinya pada absis.
2. Ambil titik minimum (sesuai absis) dan maksimum untuk dijadikan acuan pertama.
3. Bagi persoalan menjadi dua dengan menarik garis dari titik minimum ke titik maksimum.
4. Selesaikan persoalan untuk titik-titik yang berada pada bagian atas garis.
 - a. Cari titik dengan jarak terjauh dari garis awal. Sebut titik ini dengan sebutan p_{\max} dan simpan pada daftar titik konveks.

- b. Tarik garis dari titik awal menuju p_{\max} dan p_{\max} menuju titik akhir.
 - c. Selesaikan persoalan untuk titik-titik yang berada pada bagian atas kedua garis.
- 5. Selesaikan persoalan untuk titik-titik yang berada pada bagian bawah garis.
 - a. Cari titik dengan jarak terjauh dari garis awal. Sebut titik ini dengan sebutan p_{\min} dan simpan pada daftar titik konveks.
 - b. Tarik garis dari titik awal menuju p_{\min} dan p_{\min} menuju titik akhir.
 - c. Selesaikan persoalan untuk titik-titik yang berada pada bagian bawah kedua garis.
- 6. Gabungkan semua solusi yang telah ditemukan dan kembalikan nilainya.
- 7. Visualisasikan hasil yang didapat.

BAB 2

IMPLEMENTASI ALGORITMA DALAM BAHASA JAVA

Dalam pembuatan program ini, penulis menggunakan bahasa pemrograman Python.

2.1 myConvexHull.py

File ini berisi pustaka myConvexHull yang berguna untuk mencari convex hull dari kumpulan titik. Adapun pustaka ini terdiri dari beberapa fungsi, yaitu:

- **Attributes**

Function	Description
myConvexHull(bucket)	Fungsi ini bertindak sebagai fungsi utama pada pustaka yang akan mengembalikan titik-titik konveks. Masukannya berupa array of points([x,y]).
myConvexHullRecursion(p_start, p_end, table, isUpper)	Fungsi ini bertindak sebagai fungsi pembantu rekursif yang akan mencari titik-titik konveks secara rekursif.
pointDetermination(line, point)	Fungsi ini berguna untuk mengembalikan jarak antara titik dengan garis. Garis disimbolkan sebagai dua point.

2.2 main.py dan main.ipynb

File ini berisi kode program yang bertindak sebagai *driver* dan contoh untuk penggunaan pustaka myConvexHull.

2.3 Dependencies

1. numpy
2. pandas
3. matplotlib
4. sklearn

BAB 3

SOURCE CODE PROGRAM

3.1. Repository Program

Repository program dapat diakses melalui:

- Github: <https://github.com/gagaspbahar/convex-hull-2d-dnc>

3.2. Source Code Program

- 3.2.1. main.py dan main.ipynb

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from sklearn import datasets
5 from src.myConvexHull import myConvexHull
6
7 print('
8 Welcome to Convex Hull Visualization!
9 Enter your desired dataset:
10 1. Iris (from sklearn)
11 2. Wine (from sklearn)
12 3. Breast Cancer (from sklearn)
13 4. Custom dataset (make sure it has target column)
14 ')
15
16 prompt = int(input())
17
18 if prompt == 1:
19     data = datasets.load_iris()
20 elif prompt == 2:
21     data = datasets.load_wine()
22 elif prompt == 3:
23     data = datasets.load_breast_cancer()
24 elif prompt == 4:
25     filename = input("Masukkan nama file csv yang berada pada folder test: ")
26 else:
27     print("Data tidak valid. Mengembalikan hasil untuk iris..")
28     data = datasets.load_iris()
29
30 if (prompt != 4):
31     print("Masukkan pilihan kolom pertama: ")
32     for i in range(len(data.feature_names)):
33         print(str(i+1) + ". " + data.feature_names[i].title())
34     first_choice = int(input()) - 1
35
36     print("Masukkan pilihan kolom kedua: ")
37     for i in range(len(data.feature_names)):
38         print(str(i+1) + ". " + data.feature_names[i].title())
39     second_choice = int(input()) - 1
40
41 df = pd.DataFrame(data.data, columns=data.feature_names)
42 df['Target'] = pd.DataFrame(data.target)
43
44 #visualisasi hasil ConvexHull
45 plt.figure(figsize = (10, 6))
46 colors = ['b','r','g']
47 plt.title(str(data.feature_names[first_choice].title()) + " vs " + str(data.feature_names[second_choice].title()))
48 plt.xlabel(data.feature_names[first_choice])
49 plt.ylabel(data.feature_names[second_choice])
50 for i in range(len(data.target_names)):
51     bucket = df[df['Target'] == i]
52     bucket = bucket.iloc[:,[first_choice, second_choice]].values
53     hull = myConvexHull(bucket)
54     plt.scatter(bucket[:, 0], bucket[:, 1], label=data.target_names[i])
55     for j in range(len(hull)-1):
56         plt.plot([hull[j][0], hull[j+1][0]], [hull[j][1], hull[j+1][1]], colors[i])
57     plt.plot([hull[len(hull)-1][0], hull[0][0]], [hull[len(hull)-1][1], hull[0][1]], colors[i])
58 plt.legend()
59 plt.show()
60
61 else:
62     df = pd.read_csv("test/" + filename)
63     if ('target') not in df:
64         print("File tidak memiliki kolom target. Exiting..")
65         quit()
66
67     print("Masukkan pilihan kolom pertama: ")
68     for i in range(len(df.columns)-1):
69         print(str(i+1) + ". " + df.columns.values[i].title())
70     first_choice = int(input()) - 1
71
72     print("Masukkan pilihan kolom kedua: ")
73     for i in range(len(df.columns)-1):
74         print(str(i+1) + ". " + df.columns.values[i].title())
75     second_choice = int(input()) - 1
76
77 #visualisasi hasil ConvexHull
78 plt.figure(figsize = (10, 6))
79 colors = ['b','r','g']
80 plt.title(str(df.columns.values[first_choice].title()) + " vs " + str(df.columns.values[second_choice].title()))
81 plt.xlabel(df.columns.values[first_choice])
82 plt.ylabel(df.columns.values[second_choice])
83 for i in range(len(df['target'].unique())):
84     bucket = df[df['target'] == i]
85     bucket = bucket.iloc[:,[first_choice, second_choice]].values
86     hull = myConvexHull(bucket)
87     plt.scatter(bucket[:, 0], bucket[:, 1], label=df['target'].unique()[i])
88     for j in range(len(hull)-1):
89         plt.plot([hull[j][0], hull[j+1][0]], [hull[j][1], hull[j+1][1]], colors[i])
90     plt.plot([hull[len(hull)-1][0], hull[0][0]], [hull[len(hull)-1][1], hull[0][1]], colors[i])
91 plt.legend()
92 plt.show()

```


3.2.2. myConvexHull.py

```
1 import numpy as np
2 convexList = []
3
4 # Main convex hull function
5 def myConvexHull(bucket):
6     # variable init
7     sorted_bucket = bucket[bucket[:, 0].argsort()]
8     sorted_bucket = sorted_bucket.tolist()
9     p1 = sorted_bucket[0]
10    pn = sorted_bucket[-1]
11    # recurse, store memo in convexList.
12    global convexList
13    convexList = [p1, pn]
14    maxtable = [point for point in sorted_bucket if pointDetermination([p1, pn], point) > 0]
15    mintable = [point for point in sorted_bucket if pointDetermination([p1, pn], point) < 0]
16    myConvexHullRecursion(p1, pn, maxtable, True)
17    myConvexHullRecursion(p1, pn, mintable, False)
18    return convexList
19
20 # recursive function
21 def myConvexHullRecursion(p_start, p_end, table, isUpper):
22     global convexList
23     # case if its the upper part (take the max)
24     if(isUpper):
25         if(len(table) == 0):
26             # base case if 0
27             return
28         else:
29             p_start_index = convexList.index(p_start)
30             if(len(table) == 1):
31                 # Base case if 1
32                 p_max = table[0]
33                 convexList.insert(p_start_index+1, p_max)
34                 return
35             else:
36                 # convex point determination
37                 points = [pointDetermination([p_start, p_end], point) for point in table]
38                 max_index = points.index(max(points))
39                 p_max = table[max_index]
40                 convexList.insert(p_start_index+1, p_max)
41                 new_table_start = [point for point in table if pointDetermination([p_start, p_max], point) > 0]
42                 new_table_end = [point for point in table if pointDetermination([p_max, p_end], point) > 0]
43                 myConvexHullRecursion(p_start, p_max, new_table_start, True)
44                 myConvexHullRecursion(p_max, p_end, new_table_end, True)
45     else:
46         if(len(table) == 0):
47             return
48         else:
49             p_end_index = convexList.index(p_end)
50             if(len(table) == 1):
51                 p_min = table[0]
52                 convexList.insert(p_end_index+1, p_min)
53                 return
54             else:
55                 points = [pointDetermination([p_start, p_end], point) for point in table]
56                 min_index = points.index(min(points))
57                 p_min = table[min_index]
58                 convexList.insert(p_end_index+1, p_min)
59                 new_table_start = [point for point in table if pointDetermination([p_start, p_min], point) < 0]
60                 new_table_end = [point for point in table if pointDetermination([p_min, p_end], point) < 0]
61                 myConvexHullRecursion(p_start, p_min, new_table_start, False)
62                 myConvexHullRecursion(p_min, p_end, new_table_end, False)
63
64 # point determination function via determinant
65 def pointDetermination(line, point):
66     p1 = line[0]
67     p2 = line[-1]
68     a = np.array([[p1[0], p1[1], 1], [p2[0], p2[1], 1], [point[0], point[1], 1]])
69     return np.linalg.det(a)
```

BAB 4

MASUKAN DAN LUARAN PROGRAM

4.1. Interface Program

Saat menjalankan file main.py, akan tampil menu seperti dibawah ini. Untuk sekarang, terdapat tiga dataset yang disupport, yaitu Iris, Wine, dan Breast Cancer.

```
Welcome to Convex Hull Visualization!
Enter your desired dataset:
1. Iris (from sklearn)
2. Wine (from sklearn)
3. Breast Cancer (from sklearn)

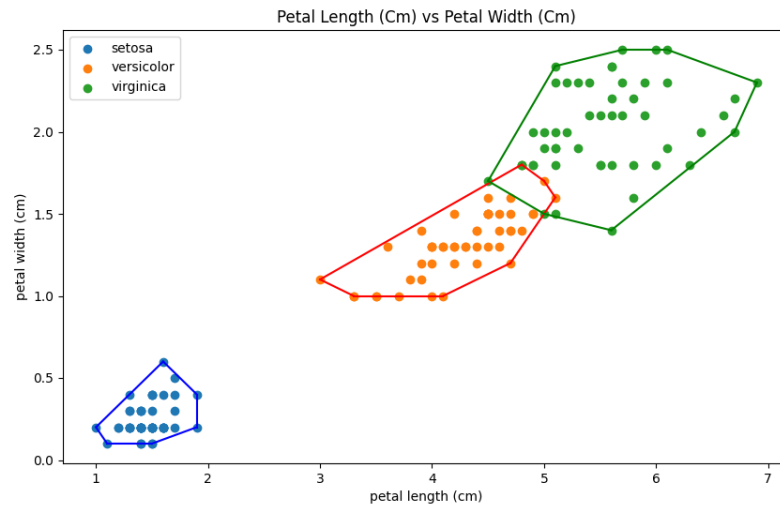
1
Masukkan pilihan kolom pertama:
1. Sepal Length (Cm)
2. Sepal Width (Cm)
3. Petal Length (Cm)
4. Petal Width (Cm)
2
Masukkan pilihan kolom kedua:
1. Sepal Length (Cm)
2. Sepal Width (Cm)
3. Petal Length (Cm)
4. Petal Width (Cm)
4
```

4.2. Dataset Iris

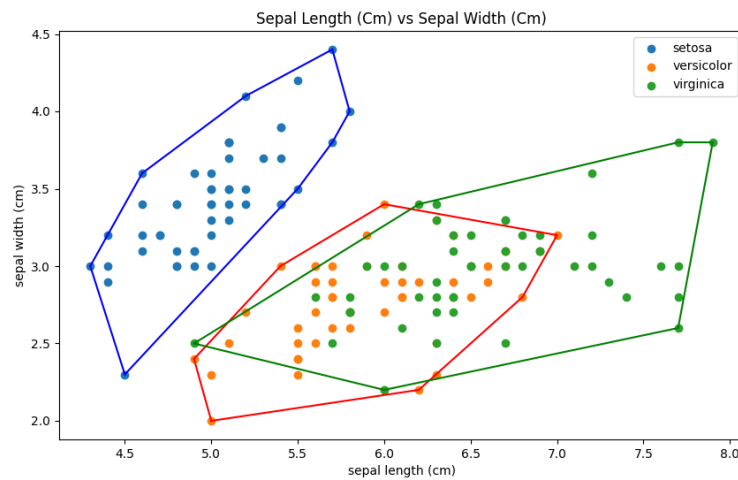
Head:

	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

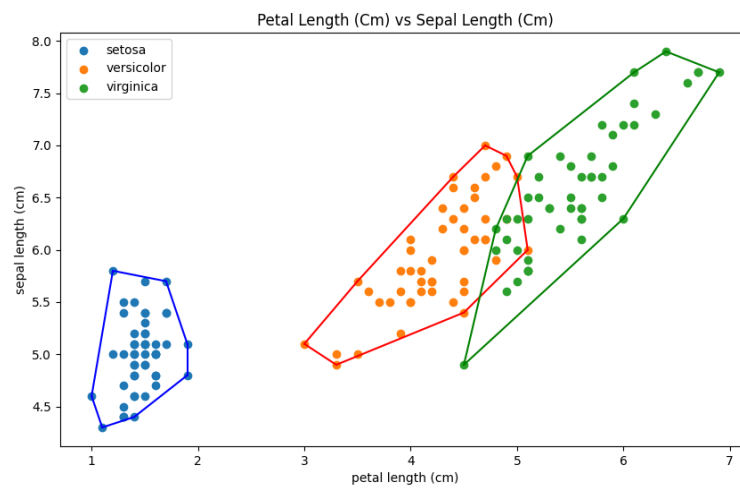
a. petal-length vs petal-width



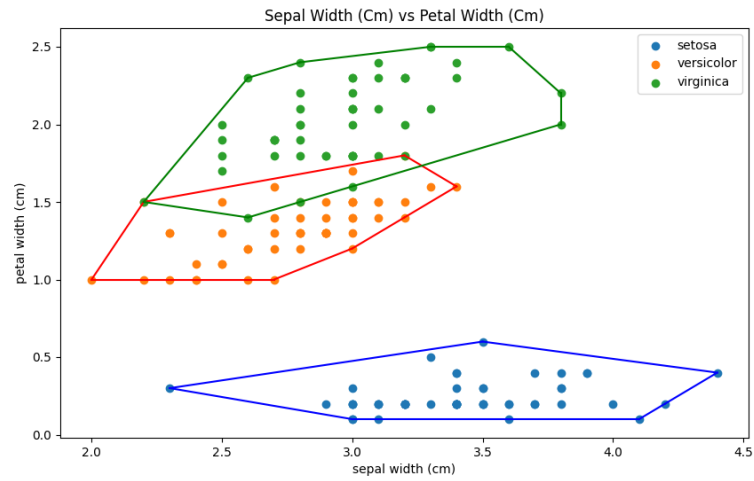
b. sepal-length vs sepal-width



c. petal-length vs sepal-length



d. sepal-width vs petal-width

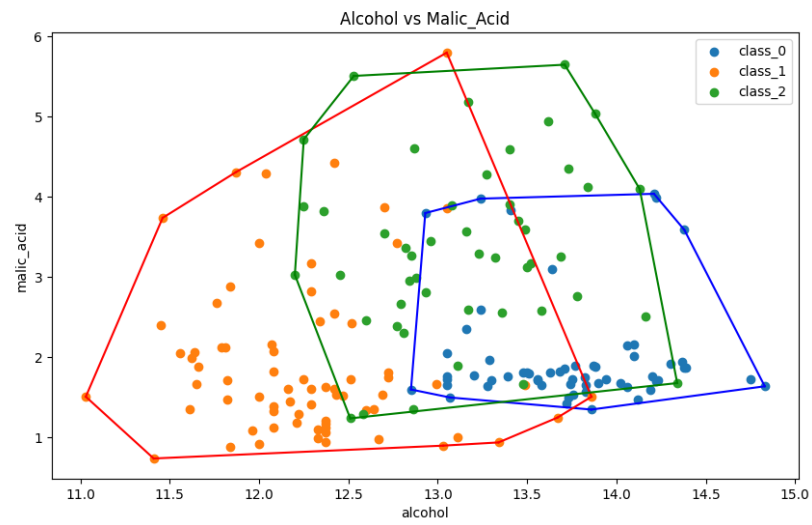


4.3. Dataset Wine

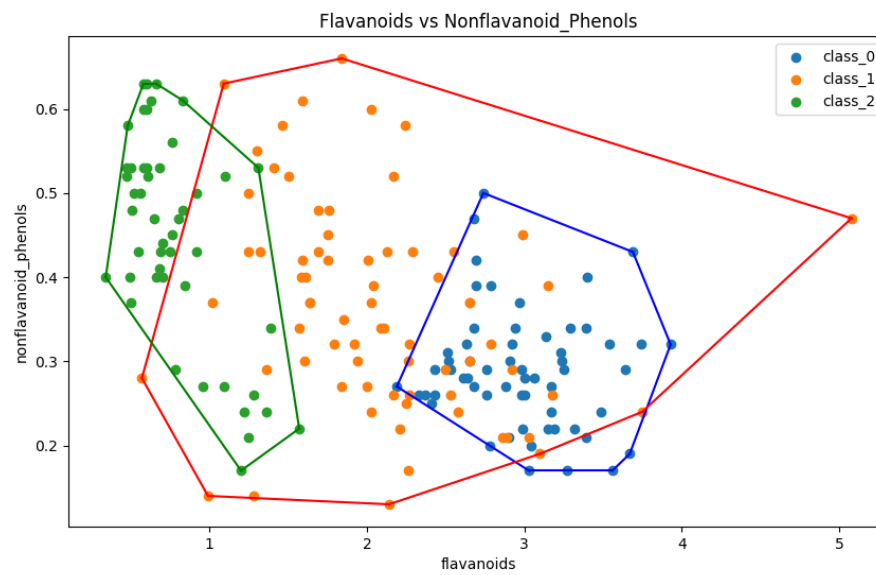
Head:

	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

a. Alcohol vs Malic Acid



b. Flavanoids vs Nonflavanoid Phenols



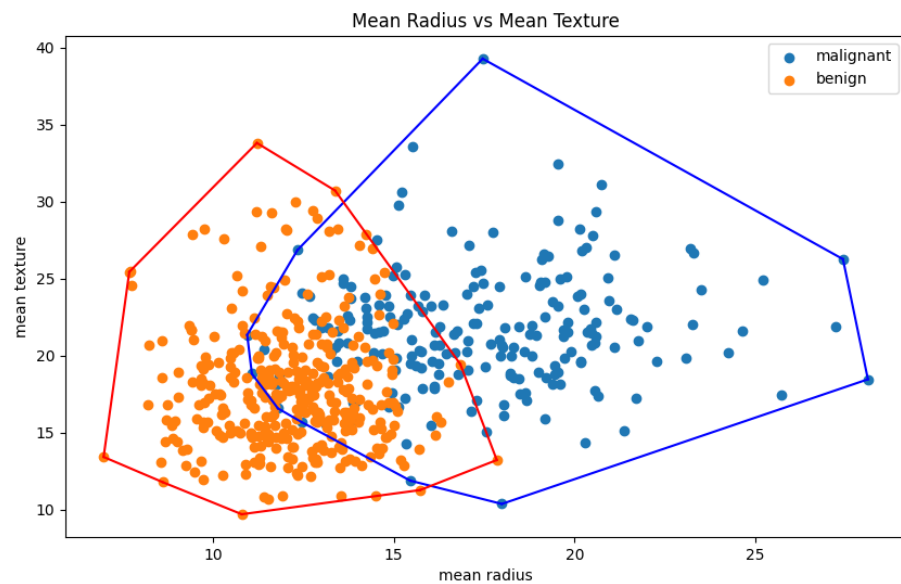
4.4. Dataset Breast Cancer

Head:

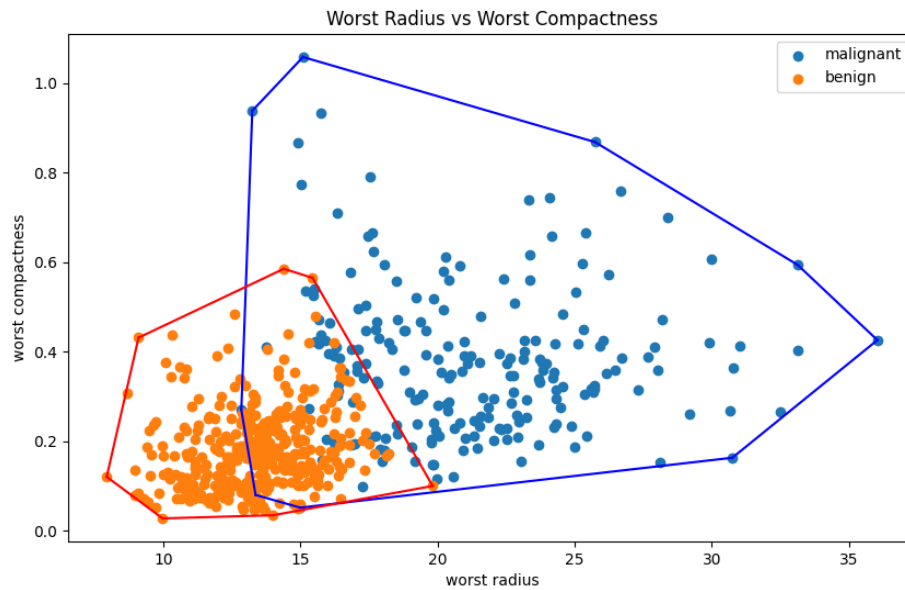
	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension	Target
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	—	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890	0
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	—	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902	0
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	—	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758	0
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	—	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300	0
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	—	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678	0

5 rows × 31 columns

a. Mean Radius vs Mean Texture



b. Worst Radius vs Worst Compactness



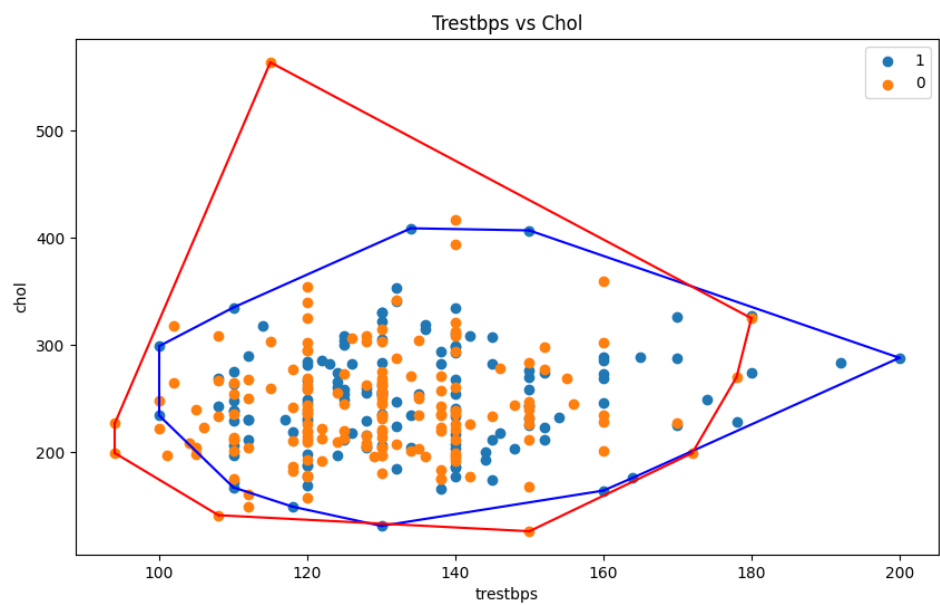
4.5. Dataset heart.csv

Head:

... (303, 14)

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

a. Trestbps vs Chol



REFERENSI

[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)

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-\(2021\)-Bagian3.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian3.pdf)

LAMPIRAN

Checklist Fitur

Poin	Ya	Tidak
1. Pustaka myConvexHull berhasil dibuat dan tidak ada kesalahan	v	
2. Convex hull yang dihasilkan sudah benar	v	
3. Pustaka myConvexHull dapat digunakan untuk menampilkan convex hull setiap label dengan warna yang berbeda.	v	
4. Bonus: program dapat menerima input dan menuliskan output untuk dataset lainnya.	v	