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:

Aira Thalca Avila Putra (K2) **13520101**

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2022

Aira Thalca Avila Putra
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

BAB I

PENJELASAN ALGORITMA PROGRAM

1.1 Divide and Conquer

Algoritma Divide and Conquer merupakan algoritma yang sangat populer di dunia Ilmu Komputer. Divide and Conquer merupakan algoritma yang berprinsip memecah-mecah permasalahan yang terlalu besar menjadi beberapa bagian kecil sehingga lebih mudah untuk diselesaikan., Solusi yang didapat dari setiap bagian kemudian digabungkan untuk membentuk sebuah solusi yang utuh. Langkah-langkah umum algoritma Divide and Conquer :

- 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

Objek masalah yang di bagi adalah masukan (input) atau instances yang berukuran n: tabel (larik), matriks, dan sebagainya, bergantung pada masalahnya. Tiap-tiap upa-masalah mempunyai karakteristik yang sama dengan karakteristik masalah asal, sehingga metode Divide and Conquer lebih natural diungkapkan dalam skema rekursif. Skema umum Algoritma Divide and Conquer :

```
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 ≤ n0 then  {ukuran persoalan P sudah cukup kecil }
     SOLVE  persoalan P yang berukuran n ini
  else
     DIVIDE  menjadi r upa-persoalan, P1, P2, …, Pr , yang masing-masing berukuran n1, n2, …, nr
     for  masing-masing P1, P2, …, Pr , do
        DIVIDEandCONQUER(Pi,  ni)
     endfor
     COMBINE  solusi dari P1, P2, …, Pr  menjadi solusi persoalan semula
  endif
```

1.2 Penjelasan Algoritma Divide dan Conquer dalam Convex Hull

Berikut langkah detail yang dilakukan dalam library yang telah dibuat :

1. Program dimulai dengan pembacaan dataset yang akan menghasilkan list yang berisi titik-titik pada setiap convex hull.
2. Dibuat list *solution* untuk menampung titik – titik yang akan menjadi convex hull
3. Dilakukan sort pada array menggunakan library python berdasarkan absis secara menaik dan jika absis bernilai sama maka diurutkan berdasarkan ordinat secara menaik
4. Setelah sort dilakukan, index pertama (p_1) dan index terakhir(p_n) dimasukkan ke array *solution*.
5. Fungsi checkPoint yang menghasilkan list berisi titik-titik digunakan untuk mengecek titik yang berada di atas atau bawah garis yang dibentuk oleh p_1 dan p_n. Titik yang berada pada garis tersebut tidak dibawa ke hasil.
6. Pengelompokkan ini dilakukan dengan membuat garis yang melalui titik p_1 dan p_n dengan persamaan $\frac{y-y_1}{(y_2-y_1)} = \frac{x-x_1}{(x_2-x_1)}$ menjadi $(x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1) = 0$ melakukan substitusi titik yang dikategorikan. Apabila hasil yang didapat adalah $(x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1) > 0$ berarti titik tersebut dimasukkan ke list Arrpos, dan sebaliknya dimasukkan ke list Arrneg.
7. Kedua list tersebut akan dimasukkan kedalam fungsi *Hull* yang menerima masukan *list of point,* titik p_1, dan titik p_n dan menghasilkan list *arr*. fungsi ini akan terus melakukan rekursif sampai masukan *list of point* adalah list kosong.
8. Pada awal fungsi hull, akan dilakukan sort sesuai nilai absolut pada fungsi pointLine yang menghitung nilai $(x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1)$ dengan x_1, y_1 adalah titik p_1 dan (x_2, y_2) adalah titik p_n. nilai x dan y adalah titik-titik yang ada pada list. Sort ini akan mengurutkan titik berdasarkan jarak titik ke garis p_1p_n secara menurun (Sebenarnya jarak titik ke garis perlu dibagi dengan panjang garis p_1p_n, namun karena panjang garis ini sama untuk semua titik, maka nilai ini tidak perlu dihitung). Masukkan titik pada index pertama (titik terjauh terhadap garis) ke list yang akan dimasukkan ke list *arr*.
9. Setelah ditemukan , akan dilakukan pengecekan untuk titik-titik pada list, jika nilai $(x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1)$ untuk titik (p_1,titik terjauh, dan p_n dan nilai persamaan yang sama untuk titik ($p_1, titik\ terjauh,$ titik yang sedang dicek) memiliki tanda yang berbeda (satu positif dan yang lainnya negative atau dengan kata lain perkalian keduanya negatif), maka dapat dipastikan titik yang dicek berada di luar garis (p_1, titik terjauh) dan jika dilakukan hal yang sama namun untuk titik p_n dan titik terjauh maka akan didapatkan juga titik yang berada di luar garis yang dibentuk dua titik tersebut.
10. Dua list yang dihasilkan dari pengecekan itu masing-masing akan dijadikan masukkan kembali ke fungsi hull seterusnya sehignga masukkan adalah list kosong. Hasil dari fungsi hull untuk dua list tersebut akan dimasukkan ke list *arr*.

11. Fungsi rekursif ini akan berakhir saat menerima array kosong, menandakan sudah tidak ada titik yang berada diatas ataupun bawah garis yang baru dibuat.
12. Seluruh titik dari fungsi hull akan dimasukkan ke list *solution.*
13. Selanjutnya dicari titik tengah dari kumpulan titik pada *solution*. Titik ini digunakan untuk menghitung besar sudut setiap titik terhadap titik tengah. Kemudian list *solution* diurutkan berdasarkan besar sudut tersebut sehingga list yang dihasilkan sudah pasti terurut (titik ke 1 membentuk garis dengan titik ke 2, titik 2 membentuk garis dengan titik 3, dan seterusnya).
14. Titik-titik tersebut kemudian di plot.

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

FUNGSI	**KEGUNAAN**
quickhull(List)	Fungsi utama yang akan mengembalikan kumpulan titik yang merupakan bagian dari convex hull
Hull(List,pointL,pointR)	Fungsi rekursif yang melakukan Divide and Conquer sampai semua titik berhasil diproses
checkPoint(List, side, pointLeft, pointRight)	Fungsi yang mengecek setiap poin apakah berada di atasa atau bawah garis
pointLine(p1,p2,p)	Menghitung jarak antara titik p dengan garis p_1p_2
searchCenter(List)	Mencari titik tengah dari seluruh titik pada List
plotting(data, type_name, x_point, y_point, target)	Fungsi yang akan melakukan plotting convexhull beserta titik-titik pada list.

BAB II

IMPLEMENTASI PROGRAM

2.1 Source Program dalam Python

2.1.1 Library myConvexHull.py

```python
from math import atan2
from matplotlib import pyplot as plt

def quickhull(List):
    answer = []
    List.sort(key = lambda arr: [arr[0], arr[1]])
    answer.append(List[0])
    answer.append(List[-1])

    Arrpos = checkPoint (List,"POS",List[0],List[-1])
    Arrneg = checkPoint (List,"NEG",List[0],List[-1])
    answer += (Hull(Arrpos,List[0],List[-1]))
    answer += (Hull(Arrneg,List[0],List[-1]))
    center = searchCenter(answer)
    answer.sort(key = lambda point: (atan2(point[1]-center[1],point[0]-center[0])))
    return answer

def checkPoint(List, side, pointLeft, pointRight):
    newList = []
    for i in range(len(List)):
        check = pointLine(pointLeft, pointRight, List[i])
        if (side == "POS" and check > 0):
            newList.append(List[i])
        elif (side == "NEG" and check < 0):
            newList.append(List[i])
    return newList

def Hull(List,pointL,pointR): #divide and conquer
    ans = []
    if (len(List) == 0):
        return ans
    List.sort(key = lambda point:abs(pointLine(pointL,pointR,point)), reverse=True)
    ans.append(List[0])
    arrpos = []
    arrneg = []
    for point in List:
        if pointLine(pointL,List[0],point) * pointLine(pointL,List[0],pointR) < 0:
            arrpos.append(point)
        if pointLine(pointR,List[0],point) * pointLine(pointR,List[0],pointL) < 0:
            arrneg.append(point)
```

```
    ans += Hull(arrpos,pointL,List[0])
    ans += Hull(arrneg,List[0],pointR)
    return ans

def pointLine(p1,p2,p):
    return (p2[0]-p1[0])*(p[1]-p1[1]) - (p2[1]-p1[1])*(p[0]-p1[0])

def searchCenter(List):
    x = y = 0
    for point in List:
        x += point[0]
        y += point[1]
    center = [x/len(List), y/len(List)]
    return center

def plotting(data, type_name, x_point, y_point, target):
    data_plot = data.copy()
    plt.figure(figsize = (10, 6))
    colors = ["blue","orange","green","red","purple","brown","pink","gray","olive","cyan"]
    plt.title(f"{x_point} vs {y_point}")
    plt.xlabel(x_point)
    plt.ylabel(y_point)
    if (target != "NONE"):
        i = 0
        for v in data_plot[target].unique():
            bucket = data_plot[data_plot[target] == v]
            datasets = bucket[[x_point,y_point]].values.tolist()
            hull = quickhull(datasets)
            ansX, ansY = [x for x in zip(*hull)]
            ansX = list(ansX)
            ansY = list(ansY)
            ansX.append(hull[0][0])
            ansY.append(hull[0][1])
            plt.scatter(bucket[x_point].values, bucket[y_point].values, label=type_name[i])
            plt.plot(list(ansX), list(ansY), colors[i%len(colors)])
            i += 1
        plt.legend()
    else:
        bucket = data_plot
        datasets = bucket[[x_point,y_point]].values.tolist()
        hull = quickhull(datasets)
        ansX, ansY = [x for x in zip(*hull)]
        ansX = list(ansX)
        ansY = list(ansY)
        ansX.append(hull[0][0])
        ansY.append(hull[0][1])
        plt.scatter(bucket[x_point].values, bucket[y_point].values)
        plt.plot(list(ansX), list(ansY), colors[2])
    plt.show()
```

2.2.1 main.py

```python
from myConvexHull import plotting
import pandas as pd
from sklearn import datasets
import os

def chooseCSV():
    print("Current Dataset:")
    dir = os.getcwd()
    dir += f"\\dataset"
    list_dataset = os.listdir(dir)
    for i in range(len(list_dataset)):
        print(f"{i+1}. {list_dataset[i]}")
    a = int(input("Masukkan data csv yang ingin dipilih (dalam angka): "))
    if a <= 0 or a > len(list_dataset):
        print("\ndata yang dipilih tidak ditemukan.\n")
        return chooseCSV()
    return f"dataset\\{list_dataset[a-1]}"

print("CONVEX HULL VISUALIZER")
print("1. iris")
print("2. wine")
print("3. breast-cancer")
print("4. manual csv")
a = int(input("Masukkan dataset yang akan dipilih: "))
while (a < 1 or a > 4):
    print("\nMasukkan salah, silahkan ulangi!\n")
    a = int(input("Masukkan dataset yang akan dipilih: "))
if a == 1:
    data = datasets.load_iris()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    target_name = data.target_names
    check = True
elif a == 2:
    data = datasets.load_wine()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    target_name = data.target_names
    check = True
elif a == 3:
    data = datasets.load_breast_cancer()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    target_name = data.target_names
    check = True
elif a == 4:
    df = chooseCSV()
    df = pd.read_csv(df)
    choose = input("Apakah dataset memiliki target? (Y/N): ")
```

```
    while (str.lower(choose) != 'y' and str.lower(choose) != 'n'):
        choose = input("Apakah dataset memiliki target? (Y/N): ")
    if (str.lower(choose) == 'y'):
        check = True
        target_name = df.target.unique()
    else:
        target_name = []
        check = False
df.head()

print("list column: ")
for i in range (len(df.columns)):
    print(f"{i+1}. {df.columns[i]}")

x_point = int(input("attribute as x (dalam angka): "))
y_point = int(input("attribute as y (dalam angka): "))
while (x_point <= 0 or x_point > len(df.columns)):
    x_point = int(input("attribute as x (dalam angka): "))
while (y_point <= 0 or y_point > len(df.columns)):
    y_point = int(input("attribute as y (dalam angka): "))
x_point = df.columns[x_point-1]
y_point = df.columns[y_point-1]

if(check):
    target = int(input("attribute as target (dalam angka):"))
    while (target <= 0 or target > len(df.columns)):
        target = int(input("attribute as target (dalam angka):"))
    target = df.columns[target-1]
else:
    target = "NONE"
try:
    plotting(df,target_name,x_point,y_point,target)
except:
    print("Data tidak bisa diplot.")
```

BAB III

PENGUJIAN PROGRAM

3.1 Input dan Output Pengujian Program

Dilakukan pengujian library convex hull untuk 5 dataset berbeda, 3 dataset berasal dari modul python scikit learn dan 2 dataset lagi berasal dari CSV. Akan diambil 2 pasangan atribut yang akan diplot dan dicari convex hullnya. Berikut hasil pengujian :

3.1.1 Interaksi awal Program

Berikut interaksi yang akan dilakukan pengguna saat memakai program

```
CONVEX HULL VISUALIZER
1. iris
2. wine
3. breast-cancer
4. manual csv
Masukkan dataset yang akan dipilih: 4
Current Dataset:
1. banknote.csv
2. heart.csv
3. indiansdiabetes.csv
Masukkan data csv yang ingin dipilih (dalam angka): 1
Apakah dataset memiliki target? (Y/N): Y
list column:
1. variance
2. skewness
3. kurtosis
4. entropy
5. target
attribute as x (dalam angka): 1
attribute as y (dalam angka): 2
attribute as target (dalam angka):5
```

3.1.2 Pengujian Dataset

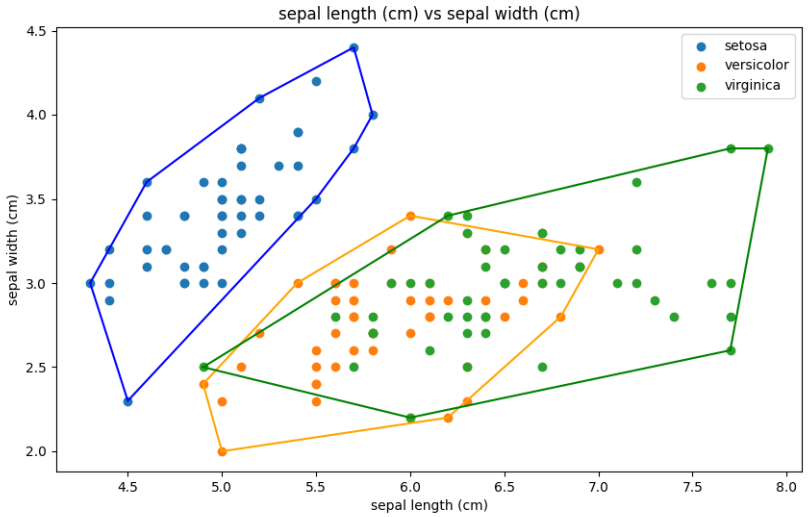
DATASET 1 : Iris

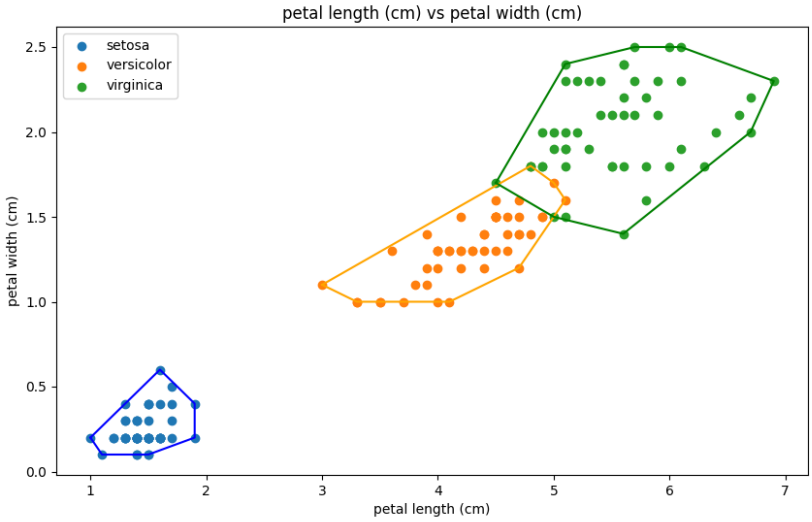
Nama Dataset :	***Pasangan Atribut :***
Iris	sepal-length, sepal-width petal-length, petal-width
Input	

Aira Thalca Avila Putra
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

	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
..	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

Output





Aira Thalca Avila Putra
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

DATASET 2 : Breast_Cancer

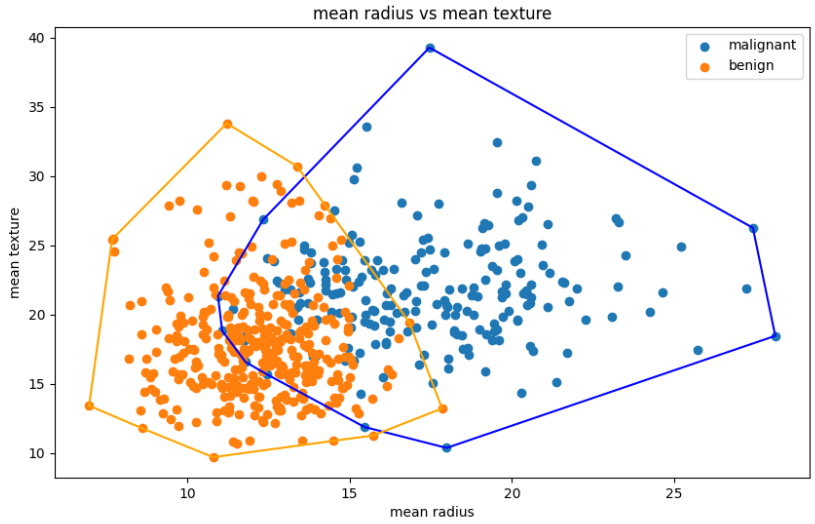
Nama Dataset :	***Pasangan Atribut :***
Breast Cancer	mean-radius, mean-texture mean-area, mean concavity

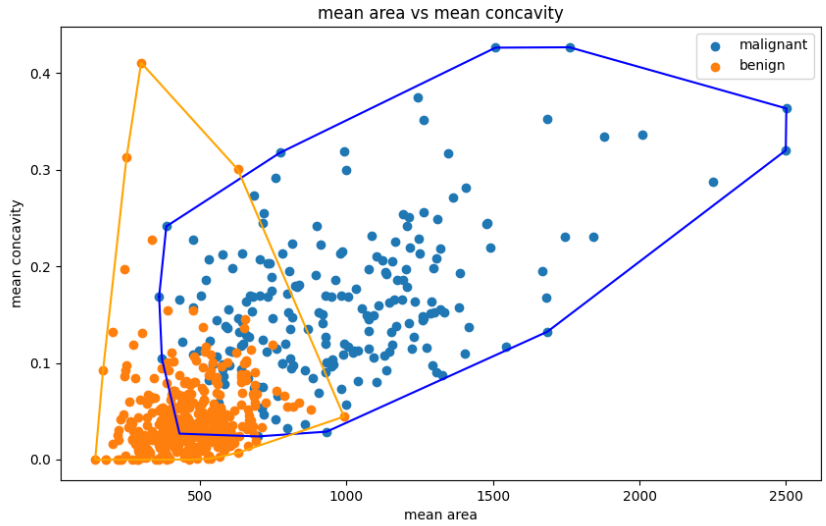
Input

	mean radius	mean texture	mean perimeter	mean area	...	worst concave points	worst symmetry	worst fractal dimension	Target
0	17.99	10.38	122.80	1001.0	...	0.2654	0.4601	0.11890	0
1	20.57	17.77	132.90	1326.0	...	0.1860	0.2750	0.08902	0
2	19.69	21.25	130.00	1203.0	...	0.2430	0.3613	0.08758	0
3	11.42	20.38	77.58	386.1	...	0.2575	0.6638	0.17300	0
4	20.29	14.34	135.10	1297.0	...	0.1625	0.2364	0.07678	0
..	...	...	...	...	...	...	...	...	...
564	21.56	22.39	142.00	1479.0	...	0.2216	0.2060	0.07115	0
565	20.13	28.25	131.20	1261.0	...	0.1628	0.2572	0.06637	0
566	16.60	28.08	108.30	858.1	...	0.1418	0.2218	0.07820	0
567	20.60	29.33	140.10	1265.0	...	0.2650	0.4087	0.12400	0
568	7.76	24.54	47.92	181.0	...	0.0000	0.2871	0.07039	1

[569 rows x 31 columns]

Output





Aira Thalca Avila Putra
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

DATASET 3 : Wine

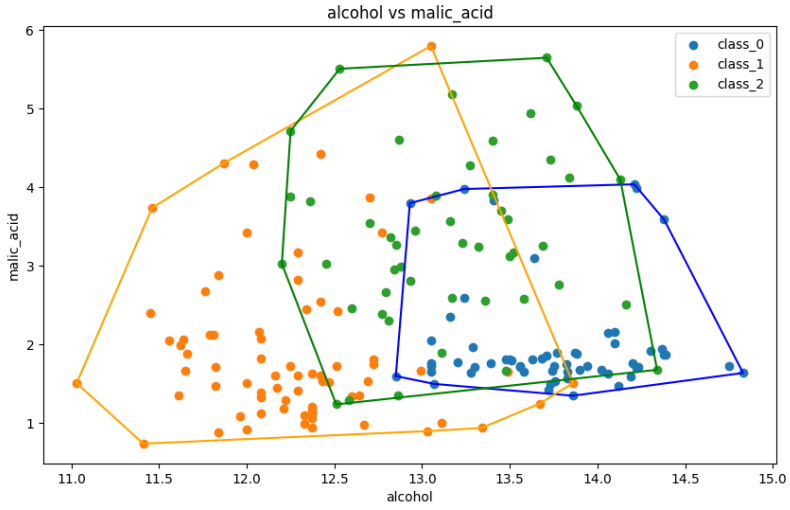
Nama Dataset :	***Pasangan Atribut :***
Wine	alcohol, malic-acid ash,magnesium

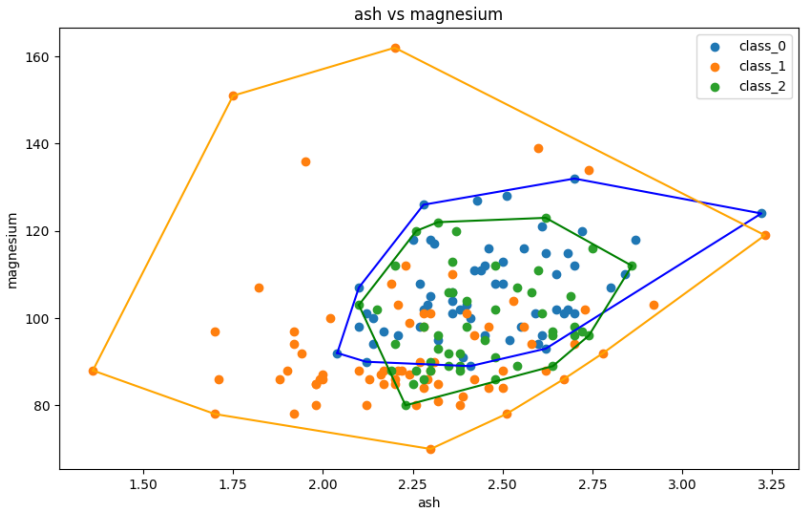
Input

	alcohol	malic_acid	ash	alcalinity_of_ash	...	hue	od280/od315_of_diluted_wines	proline	Target
0	14.23	1.71	2.43	15.6	...	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	...	1.05	3.40	1050.0	0
2	13.16	2.36	2.67	18.6	...	1.03	3.17	1185.0	0
3	14.37	1.95	2.50	16.8	...	0.86	3.45	1480.0	0
4	13.24	2.59	2.87	21.0	...	1.04	2.93	735.0	0
..	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	...	0.64	1.74	740.0	2
174	13.40	3.91	2.48	23.0	...	0.70	1.56	750.0	2
175	13.27	4.28	2.26	20.0	...	0.59	1.56	835.0	2
176	13.17	2.59	2.37	20.0	...	0.60	1.62	840.0	2
177	14.13	4.10	2.74	24.5	...	0.61	1.60	560.0	2

[178 rows x 14 columns]

Output





DATASET 4 : Banknote

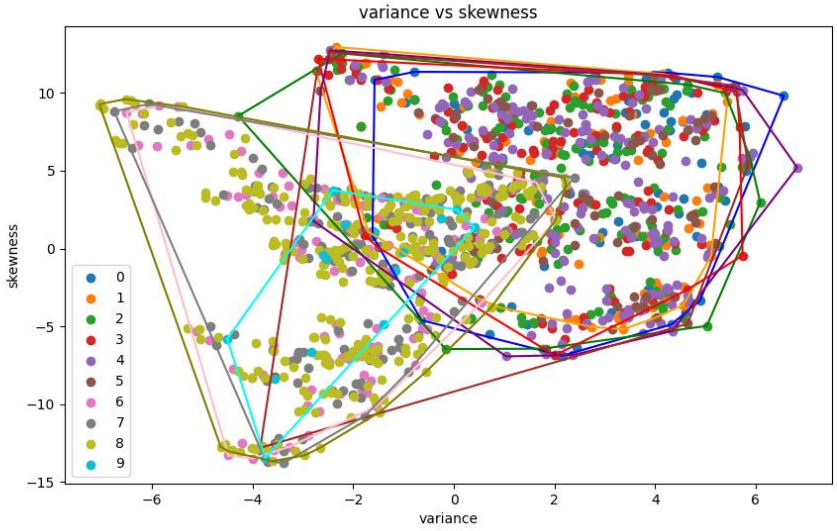
Nama Dataset :	***Pasangan Atribut :***
Banknote	variance, skewness kurtosis, entropy

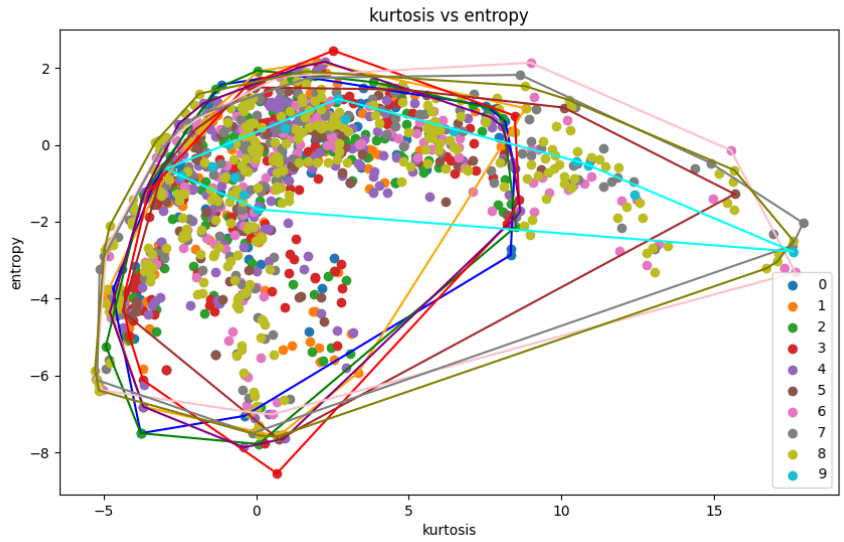
Input

```
      variance  skewness  kurtosis  entropy  target
0      3.62160   8.66610   -2.8073 -0.44699       0
1      4.54590   8.16740   -2.4586 -1.46210       0
2      3.86600  -2.63830    1.9242  0.10645       0
3      3.45660   9.52280   -4.0112 -3.59440       0
4      0.32924  -4.45520    4.5718 -0.98880       0
...        ...       ...       ...      ...     ...
1367   0.40614   1.34920   -1.4501 -0.55949       9
1368  -1.38870  -4.87730    6.4774  0.34179       9
1369  -3.75030 -13.45860   17.5932 -2.77710       9
1370  -3.56370  -8.38270   12.3930 -1.28230       9
1371  -2.54190  -0.65804    2.6842  1.19520       9

[1372 rows x 5 columns]
```

Output





DATASET 5 : Heart

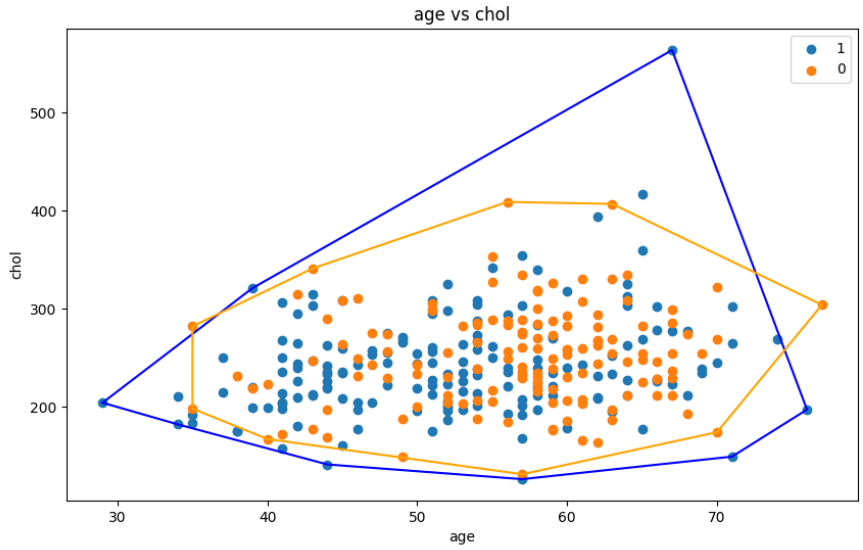
Nama Dataset :	***Pasangan Atribut :***
Heart	age, chol trestbps, exang
Input	

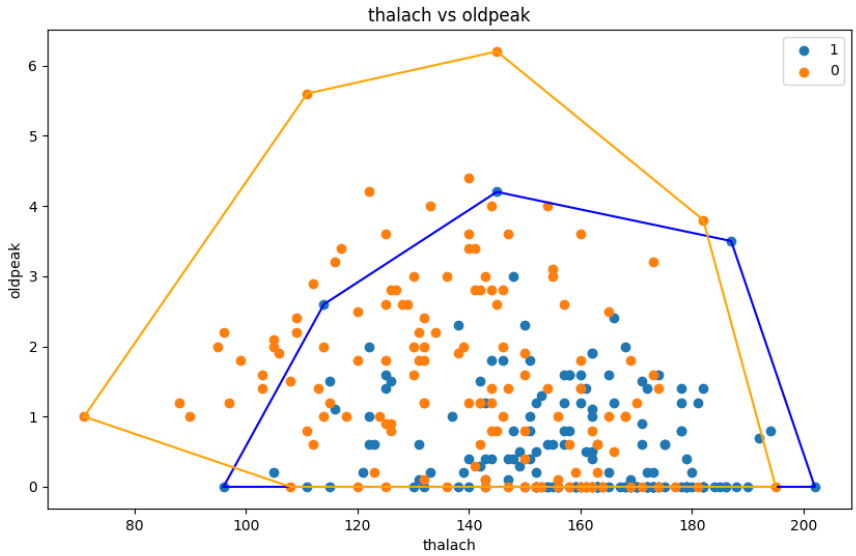
Aira Thalca Avila Putra
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

	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
..	...	...	..	...	...	...	...	...	...	...	...	..	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

[303 rows x 14 columns]

Output





Aira Thalca Avila Putra
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

LAMPIRAN

1. **REPOSITORY GITHUB :**
   https://github.com/airathalca/TUCIL-2-Strategi-Algoritma

2. **CHECKLIST:**

Poin	**Ya**	**Tidak**
1. Pustaka *myConvexHull* berhasil dibuat dan tidak ada kesalahan	✓	
2. *Convex hull* yang dihasilkan sudah benar	✓	
3. Pustaka *myConvexHull* dapat digunakan untuk menampilkan *convex hull* 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 4. Diakses pada 27 Februari 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Divide-and-Conquer-(2022)-Bagian4.pdf\
- https://andikafisma.wordpress.com/algoritma-divide-and-conquer/