

Implementasi Convex Hull untuk Visualisasi Tes *Linear Separability*

Dataset dengan Algoritma Divide and Conquer

Dibuat sebagai Tugas Kecil 2

IF2211

Strategi Algoritma



Dikerjakan oleh:

Fernaldy 13520112

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2022

DAFTAR ISI

Daftar Isi

BAB I ALGORITMA DIVIDE AND CONQUER	1
1. Class myObjectHull	1
2. Fungsi quickSort dan Partisi	1
3. Fungsi Hull.....	1
4. Fungsi det, addsimplices, dan angle.....	2
BAB II KODE PROGRAM.....	3
BAB III SKRINSUT INPUT-OUTPUT PROGRAM	10
BAB IV ALAMAT DRIVE DAN REPOSITORY	18

BAB I

ALGORITMA DIVIDE AND CONQUER

Berikut adalah garis besar algoritma *divide and conquer* yang digunakan untuk visualisasi Tes *Linear Separability Dataset*:

1. *Class myObjectHull*

Instantiasi objek dari kelas *myObjectHull* memerlukan parameter array of titik yang akan ditentukan titik-titik dari kumpulan titik tersebut yang membentuk convex hull. Array of titik tersebut akan disimpan dalam atribut *points* dalam objek. Selain atribut *points*, juga terdapat atribut *sortedidx* dan *simplices*. Atribut *sortedidx* berisi array of integer yang merupakan indeks titik pada atribut *points* apabila titik-titik tersebut diurutkan menaik berdasarkan nilai absis dan untuk nilai absis yang sama, titik akan diurutkan berdasarkan ordinat menaik. Dengan demikian, dapat diartikan bahwa elemen pertama *sortedidx* adalah indeks titik terkecil bawah dan elemen terakhir *sortedidx* adalah indeks titik terkecil atas pada *points*. Atribut *simplices* berisi array of array of integer yang mana masing-masing integer menunjukkan indeks titik pada *points* yang merupakan salah satu titik convex hull. Dengan kata lain, apabila isi *simplices* $[[0,1]]$ berarti *points*[0] dan *points*[1] merupakan pasangan titik terhubung pada convex hull.

2. *Fungsi quickSort dan Partisi*

Fungsi *quickSort* digunakan untuk melakukan pengurutan titik dengan prioritas absis menaik lalu ordinat menaik dengan memanfaatkan *divide and conquer*. Ide dasar pengurutan dengan *quickSort* adalah membagi array menjadi 2 bagian dengan bagian setiap elemen bagian pertama lebih kecil dari setiap elemen bagian kedua. Pembagian memanfaatkan fungsi *Partisi* dan terus dilakukan hingga tersisa 1 elemen. Fungsi *quickSort* dimanfaatkan untuk memperoleh atribut *sortedidx*

3. *Fungsi Hull*

Fungsi *Hull* merupakan fungsi utama yang mengimplementasikan algoritma *divide and conquer* untuk visualisasi Tes *Linear Separability Dataset*. Dalam fungsi *Hull* ini, atribut *simplices* akan diisi dengan pasangan titik yang merupakan salah satu sisi dari convex hull. Pertama, ditarik garis bayangan antara titik paling kiri bawah (p_1) dan titik paling kanan atas (p_n). Lalu titik-titik yang ada dibagi menjadi titik-titik yang memiliki determinan

positif dan negatif terhadap garis p_1p_n . Determinan positif berarti titik tersebut ada di kiri atau atas garis p_1p_n dan determinan negatif berarti titik tersebut ada di kanan atau bawah garis p_1p_n . Titik dengan determinan nol dapat diabaikan karena titik tersebut ada pada garis p_1p_n dan tidak mungkin membentuk convex hull. Kumpulan titik di atas bisa membentuk convex hull bagian atas, dan kumpulan titik pada S2 bisa membentuk convex hull bagian bawah. Untuk masing-masing kumpulan titik, apabila kumpulan titik kosong (basis rekursif), maka pasangan p_1 dan p_n ditambahkan ke dalam atribut simplices sebagai sisi yang membentuk convex hull. Apabila tidak kosong, maka dicari titik p_{max} yang memiliki jarak maksimum dari garis p_1p_n dan apabila ada titik dengan jarak sama, maka p_{max} dipilih dari titik dengan sudut $p_{max}p_1p_n$ maksimum. Selanjutnya dipanggil fungsi Hull untuk pasangan titik p_1 dan p_{max} serta untuk p_{max} dan p_n dengan parameter kumpulan titik yang ada di atas garis p_1p_n . Pada Hull p_1 dan p_{max} serta p_{max} dan p_n , titik yang terdapat dalam segitiga $p_1p_{max}p_n$ dapat diabaikan karena tidak akan membentuk convex hull. Rekursif dilakukan hingga menemukan kasus basis yaitu ketika himpunan titik di luar garis kosong yang berarti garis tersebut adalah garis yang membentuk convex hull. Hal demikian dilakukan juga untuk kumpulan garis yang ada di kanan atau bawah garis p_1p_n . Dengan demikian atribut simplices akan terisi dengan pasangan titik yang merupakan kumpulan garis pembentuk convex hull.

4. Fungsi *det*, *addsimplices*, dan *angle*

Fungsi *det*(*self*, *i*, *j*, *k*) mengembalikan nilai determinan titik dengan indeks *k* pada atribut points terhadap garis yang dibentuk oleh titik dengan indeks *i* dan *j* pada atribut points. Fungsi *addsimplices*(*self*, *a*, *b*) menambahkan pasangan indeks *a* dan *b* sebagai pasangan indeks titik pada atribut simplices. Fungsi *angle*(*self*, *i*, *j*, *k*) mengembalikan sudut antara titik dengan indeks *i*, *k*, *j* pada atribut points.

BAB II

KODE PROGRAM

File myConvexHull.py:

```
from cmath import sqrt
import numpy as np
from math import sqrt, acos

class myConvexHull:

    def __init__ (self, points):

        self.points = points
        self.simplices = []
        self.sortedidx = self.quickSort([i for i in range(len(self.points))], 0, len(self.points)-1)
        n = len(self.sortedidx)
        self.Hull(self.sortedidx[0], self.sortedidx[n-1], self.sortedidx[1:n-1], True, True)

    def quickSort(self, idx, i, j):

        idx1 = [0 for l in range(len(idx))]
        for l in range(len(idx)):
            idx1[l] = idx[l]
        if i<j:
            idx1,k = self.Partisi(idx1, i, j)
            if (j!=k or idx!=idx1):
                idx1 = self.quickSort(idx1, i, k)
            if (i!=k+1 or idx!=idx1):
                idx1 = self.quickSort(idx1, k+1, j)
        return idx1
```

```

def Partisi(self, idx, i, j):

    pivotx = self.points[idx[(i+j) // 2]][0]
    pivoty = self.points[idx[(i+j) // 2]][1]
    p = i
    q = j

    while (self.points[idx[p]][0] < pivotx) or (self.points[idx[p]][0] == pivotx and
self.points[idx[p]][1] < pivoty):
        p = p + 1
    while (self.points[idx[q]][0] > pivotx) or (self.points[idx[q]][0] == pivotx and
self.points[idx[q]][1] > pivoty):
        q = q - 1
    if p <= q:
        temp = idx[p]
        idx[p] = idx[q]
        idx[q] = temp
        p += 1
        q -= 1

    while (p<=q):
        while (self.points[idx[p]][0] < pivotx) or (self.points[idx[p]][0] == pivotx and
self.points[idx[p]][1] < pivoty):
            p = p + 1
        while (self.points[idx[q]][0] > pivotx) or (self.points[idx[q]][0] == pivotx and
self.points[idx[q]][1] > pivoty):
            q = q - 1
        if p <= q:
            temp = idx[p]
            idx[p] = idx[q]
            idx[q] = temp

```

```

    p += 1
    q -= 1

return idx,q

def Hull(self, i, j, idp, left, right): # i = idx[0] , j = idx[n] , idp = idx[1..n-1]
    if left:
        kiri = []
        for k in range(len(idp)):
            if self.det(i, j, idp[k])>0:
                kiri.append(idp[k])
        if len(kiri)==0:
            self.addsimplices(i, j)
        else:
            p1 = np.array(self.points[i])
            p2 = np.array(self.points[j])
            p3 = np.array(self.points[kiri[0]])
            maxidx = 0 # in kiri
            maxd = abs(np.cross(p2-p1,p3-p1)/np.linalg.norm(p2-p1))
            maxangle = self.angle(kiri[0], j, i)
            for k in range(1,len(kiri)):
                p3 = np.array(self.points[kiri[k]])
                p3distance = abs(np.cross(p2-p1,p3-p1)/np.linalg.norm(p2-p1))
                p3angle = self.angle(kiri[k], j, i)
                if (p3distance > maxd) or (p3distance == maxd and p3angle > maxangle):
                    maxidx = k #in kiri
                    maxd = p3distance
                    maxangle = p3angle
            newkiri = []
            for l in range(len(kiri)):
                if l!=maxidx:

```

```

        newkiri.append(kiri[l])
    self.Hull(i, kiri[maxidx], newkiri, True, False)
    self.Hull(kiri[maxidx], j, newkiri, True, False)
if right:
    kanan = []
    for k in range(len(idp)):
        if self.det(i, j, idp[k])<0:
            kanan.append(idp[k])
    if len(kanan)==0:
        self.addsimplices(i, j)
    else:
        p1 = np.array(self.points[i])
        p2 = np.array(self.points[j])
        p3 = np.array(self.points[kanan[0]])
        maxidx = 0 # in kanan
        maxd = abs(np.cross(p2-p1,p3-p1)/np.linalg.norm(p2-p1))
        maxangle = self.angle(kanan[0], j, i)
        for k in range(1,len(kanan)):
            p3 = np.array(self.points[kanan[k]])
            p3distance = abs(np.cross(p2-p1,p3-p1)/np.linalg.norm(p2-p1))
            p3angle = self.angle(kanan[k], j, i)
            if (p3distance > maxd) or (p3distance == maxd and p3angle > maxangle):
                maxidx = k #in kanan
                maxd = p3distance
                maxangle = p3angle
        newkanan = []
        for l in range(len(kanan)):
            if l!=maxidx:
                newkanan.append(kanan[l])
        self.Hull(i, kanan[maxidx], newkanan, False, True)
        self.Hull(kanan[maxidx], j, newkanan, False, True)

```



```
return
```

```
def det(self, i, j, k):  
    ret = self.points[i][0]*self.points[j][1]  
    ret += self.points[k][0]*self.points[i][1]  
    ret += self.points[j][0]*self.points[k][1]  
    ret -= self.points[k][0]*self.points[j][1]  
    ret -= self.points[j][0]*self.points[i][1]  
    ret -= self.points[i][0]*self.points[k][1]  
    return ret
```

```
def addsimplices(self, a, b): # a,b = idx[...]  
    line = []  
    line.append(a)  
    line.append(b)  
    self.simplices.append(line)
```

```
def angle(self, i, j, k): # i,j,k = idx[...] , ret angle at k  
    a = sqrt((self.points[i][0]-self.points[j][0])**2 + (self.points[i][1]-self.points[j][1])**2)  
    b = sqrt((self.points[j][0]-self.points[k][0])**2 + (self.points[j][1]-self.points[k][1])**2)  
    c = sqrt((self.points[k][0]-self.points[i][0])**2 + (self.points[k][1]-self.points[i][1])**2)  
    temp = (b**2+c**2-a**2)/(2*b*c)  
    if (round(temp,6)==1):  
        temp = 1  
    return abs(acos(temp))
```

File main.py:

```
import numpy as np  
import pandas as pd
```

```

import matplotlib.pyplot as plt

from sklearn import datasets
from myConvexHull import myConvexHull

#choose datasets
print("\nDatasets yang tersedia adalah iris, wine, dan breast_cancer")
dataname = input("Masukkan nama datasets yang ingin digunakan: ")
while (dataname!="iris" and dataname!="wine" and dataname!="breast_cancer"):
    dataname = input("Masukkan nama datasets yang tepat (iris, wine, atau breast_cancer): ")
if dataname=="iris":
    data = datasets.load_iris()
elif dataname=="wine":
    data = datasets.load_wine()
elif dataname=="breast_cancer":
    data = datasets.load_breast_cancer()

#choose x and y
print("\nBerikut adalah atribut yang bisa dijadikan komponen x dan y:")
for a in range(len(data.feature_names)):
    print(f"{a+1}. {data.feature_names[a]}")
x = int(input(f"\nMasukkan nomor atribut yang ingin digunakan sebagai komponen x: "))
while (x<=0 or x>len(data.feature_names)):
    x = int(input(f"Masukkan nomor atribut yang tepat yang ingin digunakan sebagai komponen x: "))
x -= 1
y = int(input(f"\nMasukkan nomor atribut yang ingin digunakan sebagai komponen y: "))
while (y<=0 or y>len(data.feature_names)):
    y = int(input(f"Masukkan nomor atribut yang tepat yang ingin digunakan sebagai komponen y: "))
y -= 1

```

```

#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(f'{data.feature_names[x]} vs {data.feature_names[y]}')
plt.xlabel(data.feature_names[x])
plt.ylabel(data.feature_names[y])
for i in range(len(data.target_names)):
    bucket = df[df['Target'] == i]
    bucket = bucket.iloc[:,[x,y]].values
    hull = myConvexHull(bucket)
    plt.scatter(bucket[:, 0], bucket[:, 1], label=data.target_names[i])
    for simplex in hull.simplices:
        plt.plot(bucket[simplex, 0], bucket[simplex, 1], colors[i])
plt.legend()
plt.show()

```

BAB III

SKRINSUT INPUT-OUTPUT PROGRAM

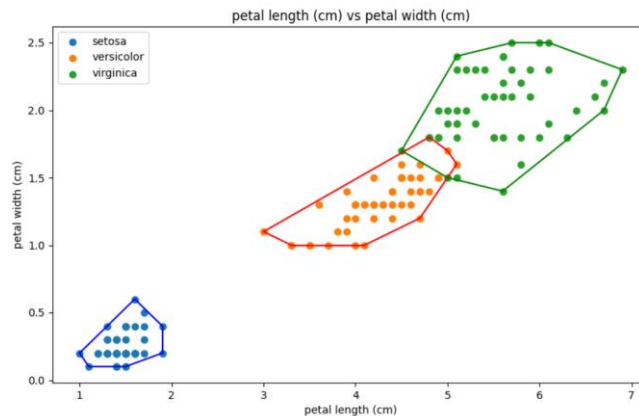
Berikut adalah skrinsut input-output program. Dataset yang bisa digunakan adalah iris, wine, dan breast_cancer.

1. Dataset iris, petal-length vs petal-width

```
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: iris

Berikut adalah atribut yang bisa dijadikan komponen x dan y:
1. sepal length (cm)
2. sepal width (cm)
3. petal length (cm)
4. petal width (cm)

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 3
Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 4
```

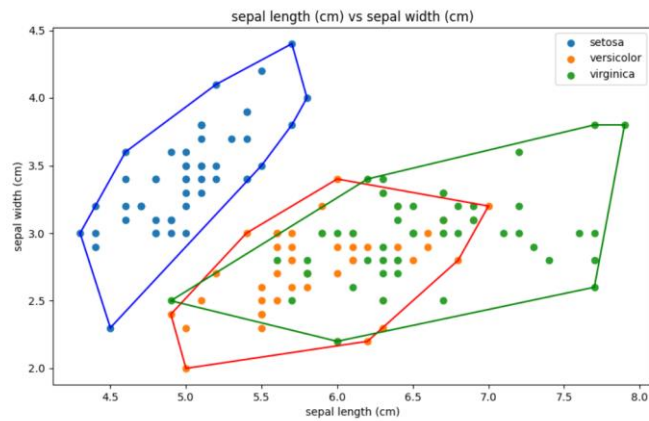


2. Dataset iris, sepal-length vs sepal-width

```
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: iris

Berikut adalah atribut yang bisa dijadikan komponen x dan y:
1. sepal length (cm)
2. sepal width (cm)
3. petal length (cm)
4. petal width (cm)

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 1
Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 2
```



3. Dataset wine, malic_acid vs ash

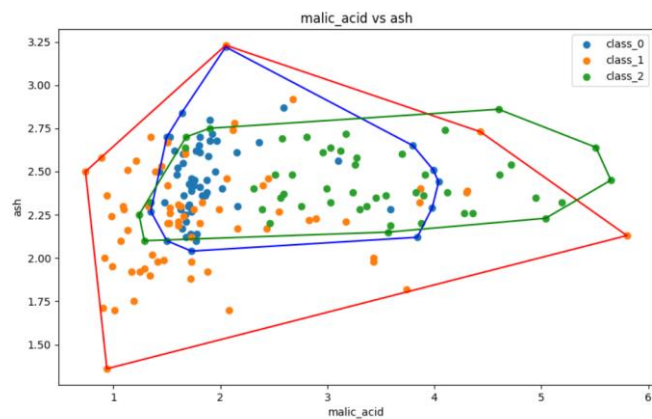
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: wine

Berikut adalah atribut yang bisa dijadikan komponen x dan y:

1. alcohol
2. malic_acid
3. ash
4. alcalinity_of_ash
5. magnesium
6. total_phenols
7. flavanoids
8. nonflavanoid_phenols
9. proanthocyanins
10. color_intensity
11. hue
12. od280/od315_of_diluted_wines
13. proline

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 2

Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 3

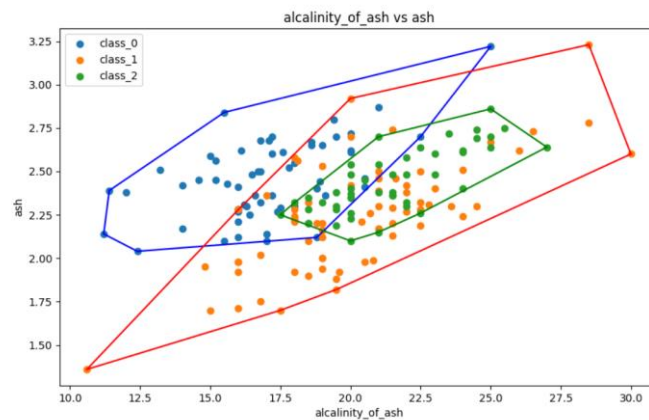


4. Dataset wine, alcalinity_of_ash vs ash

```
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: wine

Berikut adalah atribut yang bisa dijadikan komponen x dan y:
1. alcohol
2. malic_acid
3. ash
4. alcalinity_of_ash
5. magnesium
6. total_phenols
7. flavanoids
8. nonflavanoid_phenols
9. proanthocyanins
10. color_intensity
11. hue
12. od280/od315_of_diluted_wines
13. proline

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 4
Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 3
```

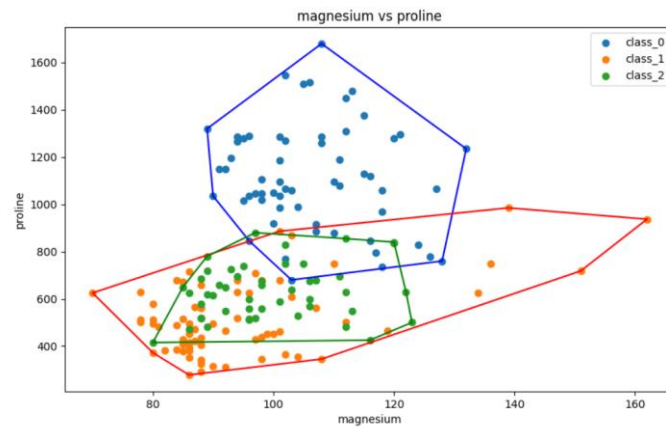


5. Dataset wine, magnesium vs proline

```
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: wine

Berikut adalah atribut yang bisa dijadikan komponen x dan y:
1. alcohol
2. malic_acid
3. ash
4. alcalinity_of_ash
5. magnesium
6. total_phenols
7. flavanoids
8. nonflavanoid_phenols
9. proanthocyanins
10. color_intensity
11. hue
12. od280/od315_of_diluted_wines
13. proline

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 5
Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 13
```



6. Dataset wine, flavanoids vs nonflavanoid_phenols

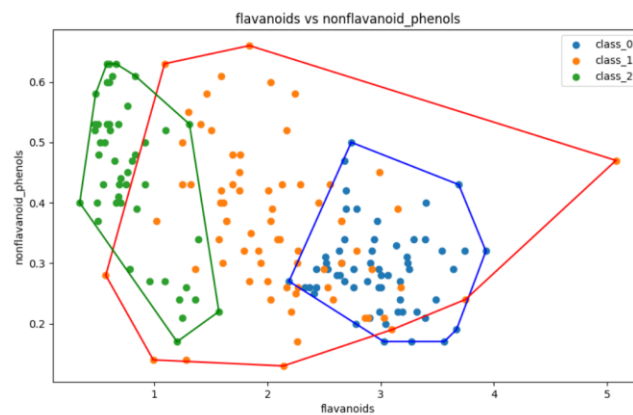
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: wine

Berikut adalah atribut yang bisa dijadikan komponen x dan y:

1. alcohol
2. malic_acid
3. ash
4. alcalinity_of_ash
5. magnesium
6. total_phenols
7. flavanoids
8. nonflavanoid_phenols
9. proanthocyanins
10. color_intensity
11. hue
12. od280/od315_of_diluted_wines
13. proline

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 7

Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 8

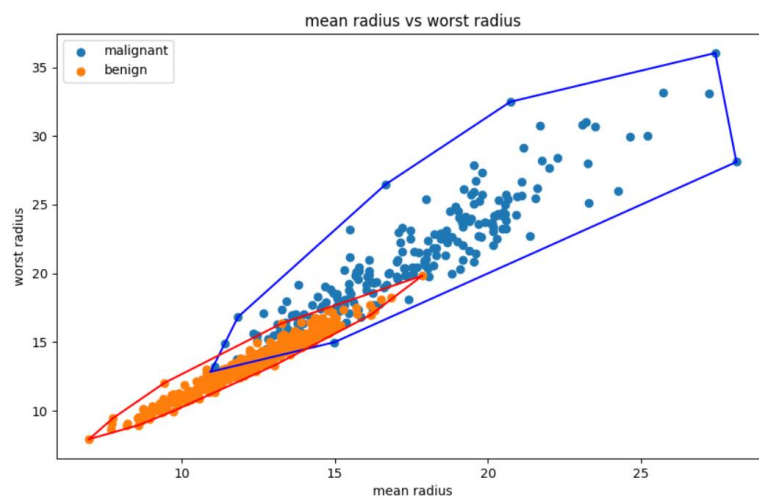


7. Dataset breast_cancer, mean radius vs worst radius

```
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: breast_cancer

Berikut adalah atribut yang bisa dijadikan komponen x dan y:
1. mean radius
2. mean texture
3. mean perimeter
4. mean area
5. mean smoothness
6. mean compactness
7. mean concavity
8. mean concave points
9. mean symmetry
10. mean fractal dimension
11. radius error
12. texture error
13. perimeter error
14. area error
15. smoothness error
16. compactness error
17. concavity error
18. concave points error
19. symmetry error
20. fractal dimension error
21. worst radius
22. worst texture
23. worst perimeter
24. worst area
25. worst smoothness
26. worst compactness
27. worst concavity
28. worst concave points
29. worst symmetry
30. worst fractal dimension

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 1
Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 21
```

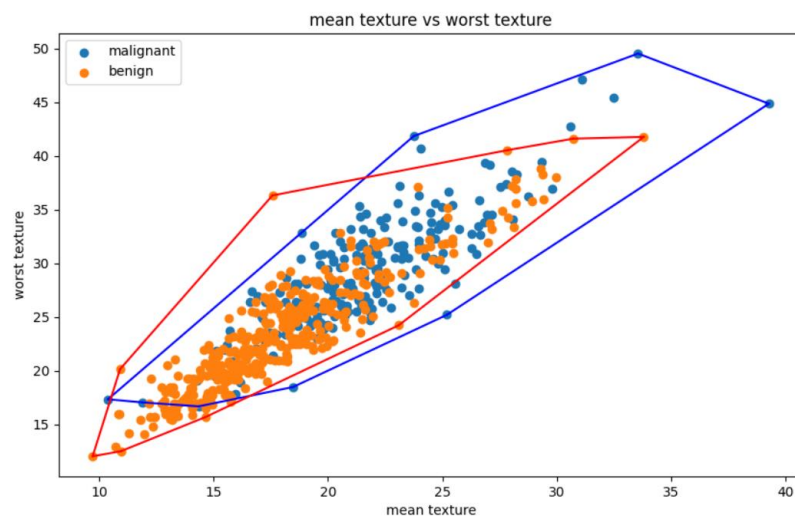


8. Dataset breast_cancer, mean texture vs worst texture

```
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: breast_cancer

Berikut adalah atribut yang bisa dijadikan komponen x dan y:
1. mean radius
2. mean texture
3. mean perimeter
4. mean area
5. mean smoothness
6. mean compactness
7. mean concavity
8. mean concave points
9. mean symmetry
10. mean fractal dimension
11. radius error
12. texture error
13. perimeter error
14. area error
15. smoothness error
16. compactness error
17. concavity error
18. concave points error
19. symmetry error
20. fractal dimension error
21. worst radius
22. worst texture
23. worst perimeter
24. worst area
25. worst smoothness
26. worst compactness
27. worst concavity
28. worst concave points
29. worst symmetry
30. worst fractal dimension

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 2
Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 22
```

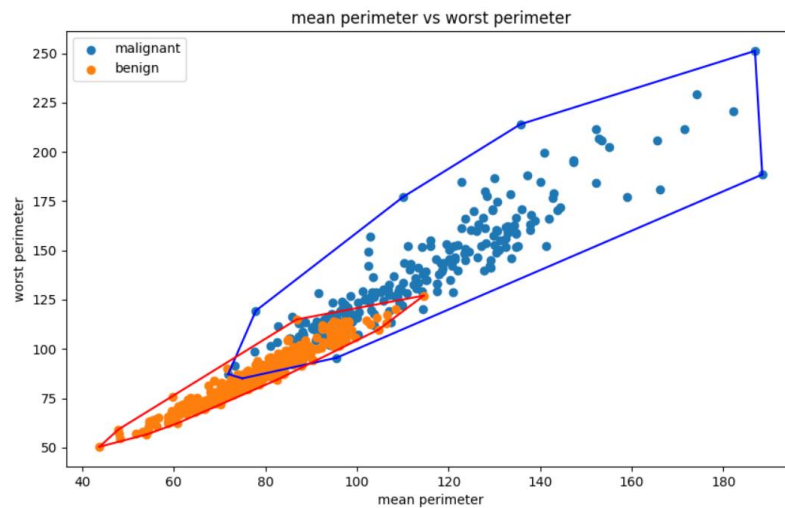


9. Dataset breast_cancer, mean perimeter vs worst perimeter

```
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: breast_cancer

Berikut adalah atribut yang bisa dijadikan komponen x dan y:
1. mean radius
2. mean texture
3. mean perimeter
4. mean area
5. mean smoothness
6. mean compactness
7. mean concavity
8. mean concave points
9. mean symmetry
10. mean fractal dimension
11. radius error
12. texture error
13. perimeter error
14. area error
15. smoothness error
16. compactness error
17. concavity error
18. concave points error
19. symmetry error
20. fractal dimension error
21. worst radius
22. worst texture
23. worst perimeter
24. worst area
25. worst smoothness
26. worst compactness
27. worst concavity
28. worst concave points
29. worst symmetry
30. worst fractal dimension

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 3
Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 23
```

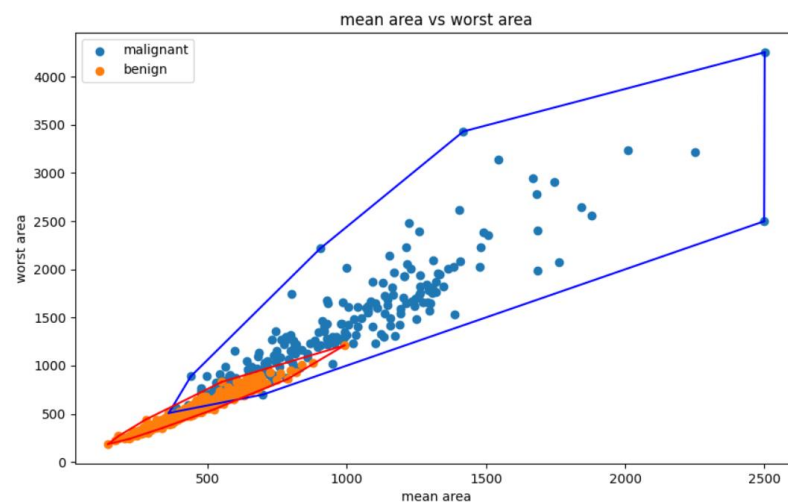


10. Dataset breast_cancer, mean area vs worst area

```
Datasets yang tersedia adalah iris, wine, dan breast_cancer
Masukkan nama datasets yang ingin digunakan: breast_cancer

Berikut adalah atribut yang bisa dijadikan komponen x dan y:
1. mean radius
2. mean texture
3. mean perimeter
4. mean area
5. mean smoothness
6. mean compactness
7. mean concavity
8. mean concave points
9. mean symmetry
10. mean fractal dimension
11. radius error
12. texture error
13. perimeter error
14. area error
15. smoothness error
16. compactness error
17. concavity error
18. concave points error
19. symmetry error
20. fractal dimension error
21. worst radius
22. worst texture
23. worst perimeter
24. worst area
25. worst smoothness
26. worst compactness
27. worst concavity
28. worst concave points
29. worst symmetry
30. worst fractal dimension

Masukkan nomor atribut yang ingin digunakan sebagai komponen x: 4
Masukkan nomor atribut yang ingin digunakan sebagai komponen y: 24
```



BAB IV

ALAMAT DRIVE DAN REPOSITORY

Alamat drive berisi source code:

<https://drive.google.com/drive/folders/1wgFodhn0AkDwNatvFCMHALQYjVQibtIh?usp=sharing>

Alamat github:

https://github.com/fernaldy112/Tucil2_13520112

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