

LAPORAN TUGAS KECIL 1
IF2211 STRATEGI ALGORITMA
CONVEX HULL



Oleh

Primanda Adyatma Hafiz (13520022)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022

Daftar Isi

Daftar Isi	1
BAB I ALGORITMA DIVIDE AND CONQUER	2
1.1 Cara Kerja Program	2
BAB 2 SOLUSI PROGRAM.....	3
2.1 Kode Program	3
BAB 3 HASIL PENGUJIAN.....	9
3.1 Sepal length vs Sepal width (Iris)	9
3.2 Petal length vs Petal Width (Iris).....	9
3.3 Alcohol vs Malic_acid (Wine)	9
3.4 Ash vs Alcalinity_of_ash (Wine).....	10
3.5 Mean radius vs Mean texture (Breast_cancer).....	10
3.6 Mean perimeter vs Mean area	10
LAMPIRAN.....	11
Link ke Repository Github	11
Tabel Penilaian	11

BAB I

ALGORITMA DIVIDE AND CONQUER

1.1 Cara Kerja Program

- 1) Mencari dua buah titik yaitu titik yang memiliki absis minimum dan titik yang memiliki absis maksimum
- 2) Hubungkan kedua titik tersebut kemudian cari titik-titik pembentuk Convex Hull pada dua buah orientasi yaitu atas dan bawah
- 3) Pencarian titik ini dilakukan secara rekursif dengan memanggil fungsi solve
- 4) Jika orientasi bernilai positif maka akan diambil seluruh titik yang memiliki orientasi positif relatif terhadap garis, sedangkan jika orientasi bernilai negatif maka akan diambil seluruh titik yang memiliki orientasi negatif relatif terhadap garis
- 5) Seluruh titik yang memenuhi syarat tersebut dimasukkan ke dalam array kemudian dicari sebuah titik yang memiliki nilai jarak terbesar terhadap garis, jika terdapat dua buah titik yang jaraknya sama akan diambil titik yang membentuk sudut terbesar
- 6) Setelah diperoleh titik yang memiliki jarak terbesar, seluruh titik yang berada diluar segitiga yang dibentuk oleh titik dengan jarak terbesar beserta dua buah titik pembentuk garis akan dimasukkan ke dalam array yang baru, sedangkan yang berada di dalam segitiga akan dibiarkan
- 7) Panggil dua buah fungsi solve untuk titik 1 pembentuk garis dan titik dengan jarak terbesar serta titik 2 pembentuk garis dan titik dengan jarak terbesar
- 8) Fungsi solve akan mencapai basis *do nothing* bila pada langkah 4 tidak terdapat titik lagi yang memenuhi
- 9) Setelah keseluruhan titik pembentuk Convex Hull diperoleh, diambil satu buah titik pivot yang berada di dalam bidang Convex Hull kemudian seluruh titik pembentuk Convex Hull diurutkan berdasarkan sudut yang dibentuk titik tersebut dengan menganggap titik pivot sebagai origin
- 10) Hubungkan titik-titik pada array yang saling bertetangga serta titik yang memiliki indeks minimum dan maksimum
- 11) Tampilkan plot garis pembentuk Convex Hull

BAB 2

SOLUSI PROGRAM

2.1 Kode Program

Berikut adalah implementasi program Convex Hull dalam bahasa Python

1) Inisialisasi Dataset

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
import matplotlib.pyplot as plt
from scipy.spatial import ConvexHull
import math
from functools import cmp_to_key

# Inisialisasi datasets
data=datasets.load_iris()
inp=0

while(not(1<=inp<=3)):
    print("List dataset :")
    print("1. Iris")
    print("2. Wine")
    print("3. Breast_cancer")
    inp=int(input("Masukkan pilihan dataset : "))
    if(inp==1):
        data = datasets.load_iris()
    elif(inp==2):
        data=datasets.load_wine()
    elif(inp==3):
        data=datasets.load_breast_cancer()
    else:
        print("Input tidak valid, silahkan ulangi")
df = pd.DataFrame(data.data, columns=data.feature_names)
df['Target'] = pd.DataFrame(data.target)

param1=0
param2=0
print("List kolom : ")
for i in range(len(data.feature_names)):
    print(str(i+1)+".",data.feature_names[i])
```

```

param1=int(input("Masukkan pilihan parameter 1 : "))
param2=int(input("Masukkan pilihan parameter 2 : "))
param1-=1
param2-=1

```

2) Hasil Visualisasi Convex Hull dari Library Scipy

```

plt.figure(figsize = (10, 6))
colors = ['b','r','g']
plt.title("Hasil library scipy")
plt.xlabel(data.feature_names[param1])
plt.ylabel(data.feature_names[param2])
for i in range(len(data.target_names)):
    bucket = df[df['Target'] == i]
    bucket = bucket.iloc[:,[param1,param2]].values
    hull = ConvexHull(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()

```

3) Realisasi Fungsi dan Prosedur serta inisialisasi variabel global

```

# Inisialisasi variabel global
listpoint=[]
bucket=[]
pivot=[0,0]
prio=[]

# Fungsi untuk menentukan orientasi p3 terhadap garis p1p2
def det(p1,p2,p3):
    return p1[0]*p2[1]+p3[0]*p1[1]+p2[0]*p3[1]-p3[0]*p2[1]-p2[0]*p1[1]-
p1[0]*p3[1]

# Fungsi untuk menghitung jarak titik p3 ke garis p1p2
def dist(p1,p2,p3):
    a=((p3[0]-p1[0])**2+(p3[1]-p1[1])**2)**0.5
    b=((p2[0]-p1[0])**2+(p2[1]-p1[1])**2)**0.5
    c=((p3[0]-p2[0])**2+(p3[1]-p2[1])**2)**0.5
    s=(a+b+c)/2
    return ((s*(s-a)*(s-b)*(s-c))**0.5)*2/b

# Fungsi untuk menghitung sudut pada p1 dari segitiga p1p2p3
def angle(p1,p2,p3):
    a=((p3[0]-p1[0])**2+(p3[1]-p1[1])**2)**0.5

```

```

b=((p2[0]-p1[0])**2+(p2[1]-p1[1])**2)**0.5
c=((p3[0]-p2[0])**2+(p3[1]-p2[1])**2)**0.5
if((a**2+b**2-c**2)/(2*a*b)>1):
    return math.acos(1)
elif((a**2+b**2-c**2)/(2*a*b)<-1):
    return math.acos(-1)
return math.acos((a**2+b**2-c**2)/(2*a*b))

# Fungsi untuk mencari p3 yang memiliki jarak maksimum dari garis p1p2
# Jika terdapat beberapa titik yang jaraknya sama akan diambil yang memiliki
sudut terbesar
def cmp(p1,p2,p3,curdist,curang):
    dis=dist(p1,p2,p3)
    ang=angle(p3,p1,p2)
    if(dis>curdist or (dis==curdist and ang>curang)):
        return True
    else:
        return False

# Fungsi untuk mengecek apakah sebuah titik p berada di dalam segitiga p1p2p3
def insideTriangle(p1,p2,p3,p):
    return not((det(p1,p2,p)>0 or det(p2,p3,p)>0 or det(p3,p1,p)>0) and
(det(p1,p2,p)<0 or det(p2,p3,p)<0 or det(p3,p1,p)<0))

# Fungsi untuk menghitung sudut dari p2 relatif terhadap origin p1
def getangle(p1,p2):
    dx=p2[0]-p1[0]
    dy=p2[1]-p1[1]
    if(dx==0):
        if(dy>0):
            return math.pi/2
        else:
            return 3*math.pi/2
    elif(dx>0 and dy>=0):
        return math.atan(dy/dx)
    elif(dx<0 and dy>=0):
        return math.atan(dy/dx)+math.pi
    elif(dx>0 and dy<0):
        return math.atan(dy/dx)+math.pi*2
    else:
        return math.atan(dy/dx)+math.pi

# Fungsi sebagai parameter untuk mengurutkan titik pada convex hull
berdasarkan besar sudutnya terhadap pivot
def getprio(a,b):
    global pivot,bucket

```

```

    return getangle(pivot,bucket[a])-getangle(pivot,bucket[b])

# Fungsi yang digunakan untuk Divide and Conquer permasalahan
def solve(arr,c1,c2,orientation):
    global bucket,listpoint
    nextarr=[]

    # Masukkan titik yang orientasinya sama dengan orientation
    for i in range(len(arr)):
        if(arr[i]==c1 or arr[i]==c2):
            continue
        if(det(bucket[c1],bucket[c2],bucket[arr[i]])<0 and orientation<0):
            nextarr.append(arr[i])
        elif(det(bucket[c1],bucket[c2],bucket[arr[i]])>0 and orientation>0):
            nextarr.append(arr[i])

    # Basis terjadi jika tidak ada titik yang memenuhi lagi
    if(len(nextarr)==0):
        return

    # Cari titik yang memiliki jarak maksimum terhadap garis c1c2
    nextpoint=nextarr[0]
    curdist=dist(bucket[c1],bucket[c2],bucket[nextpoint])
    curang=angle(bucket[c1],bucket[c2],bucket[nextpoint])
    for i in range(1,len(nextarr)):
        if(cmp(bucket[c1],bucket[c2],bucket[nextarr[i]],curdist,curang)):
            nextpoint=nextarr[i]
            curdist=dist(bucket[c1],bucket[c2],bucket[nextpoint])
            curang=angle(bucket[c1],bucket[c2],bucket[nextpoint])

    passarr=[]
    # Untuk pemanggilan fungsi rekursi selanjutnya, ambil seluruh titik yang
    berada di
    # luar segitiga c1c2nextpoint
    for i in range(len(nextarr)):
        if(not(insideTriangle(bucket[c1],bucket[c2],bucket[nextpoint],bucket[nexta
rr[i]]))):
            passarr.append(nextarr[i])
    # Catat nextpoint pada array listpoint dan lakukan proses rekursi
    selanjutnya
    listpoint.append(nextpoint)
    solve(passarr,c1,nextpoint,-det(bucket[c1],bucket[nextpoint],bucket[c2]))
    solve(passarr,nextpoint,c2,-det(bucket[nextpoint],bucket[c2],bucket[c1]))

def myConvexHull():
    global listpoint,bucket,pivot

```

```

mnmm=0
mxm=0

# Cari titik yang memiliki absis minimum dan titik yang memiliki absis
maksimums
for i in range(1,len(bucket)):
    if(bucket[mnm][0]<bucket[i][0]):
        mnm=i
    if(bucket[mxm][0]>bucket[i][0]):
        mxm=i
listpoint.append(mnm)
listpoint.append(mxm)

a=[i for i in range(len(bucket))]
b=[i for i in range(len(bucket))]

# Lakukan pencarian titik convex hull untuk kedua orientasi dari garis mnm
mxm
solve(a,mnm,mxm,1)
solve(b,mnm,mxm,-1)

res=[]
listpoint=list(dict.fromkeys(listpoint))
pivot=((bucket[mnm][0]+bucket[mxm][0])/2,(bucket[mnm][1]+bucket[mxm][1])/2)
prio=[getangle(pivot,bucket[listpoint[i]]) for i in range(len(listpoint))]

# Urutkan titik convex hull berdasarkan sudutnya terhadap pivot
listpoint=sorted(listpoint,key=cmp_to_key(getprio))

# Hubungkan titik yang saling bertetangga serta titik pertama dan terakhir
pada convex hull
for i in range(1,len(listpoint)):
    res.append([listpoint[i-1],listpoint[i]])
res.append([listpoint[len(listpoint)-1],listpoint[0]])
return res

```

4) Hasil Visualisasi Convex Hull dari Program Utama

```

# Visualisasi hasil convex hull dari program yang dibuat
plt.figure(figsize = (10, 6))
colors = ['b','r','g']
plt.title("Hasil dari Program")
plt.xlabel(data.feature_names[param1])
plt.ylabel(data.feature_names[param2])
for i in range(len(data.target_names)):
    bucket = df[df['Target'] == i]

```

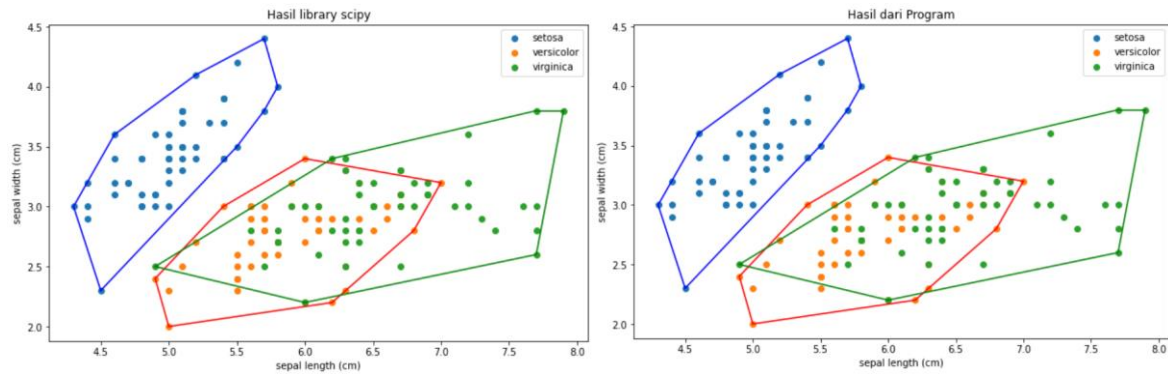


```
bucket = bucket.iloc[:,[param1,param2]].values
hull = myConvexHull()
prio=[]
listpoint=[]
plt.scatter(bucket[:, 0], bucket[:, 1], label=data.target_names[i])
for simplex in hull:
    plt.plot(bucket[simplex, 0], bucket[simplex, 1], colors[i])
plt.legend()
plt.show()
```

BAB 3 HASIL PENGUJIAN

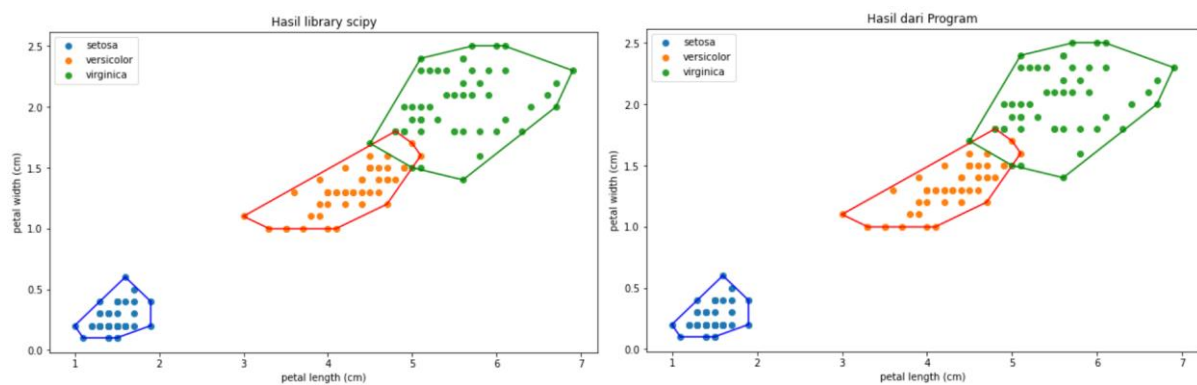
3.1 Sepal length vs Sepal width (Iris)

Masukan dataset = 1, parameter 1 = 1, parameter 2 = 2



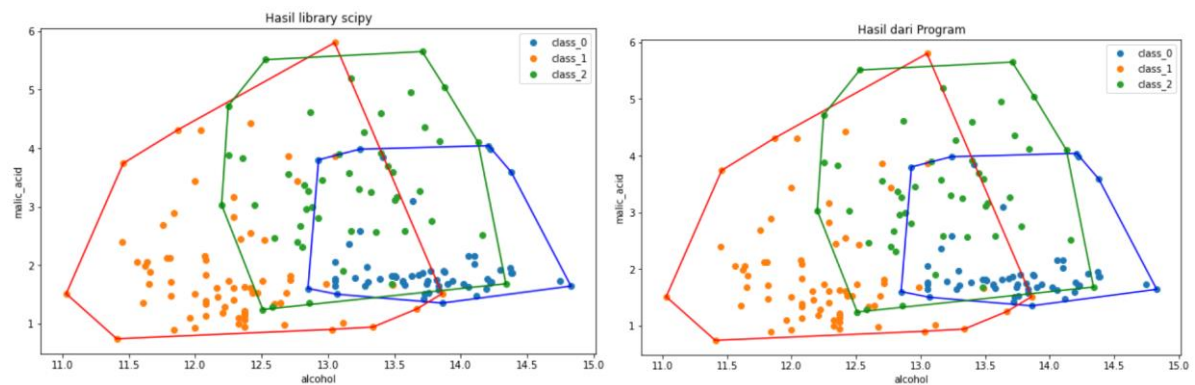
3.2 Petal length vs Petal Width (Iris)

Masukan dataset = 1, parameter 1 = 3, parameter 2 = 4



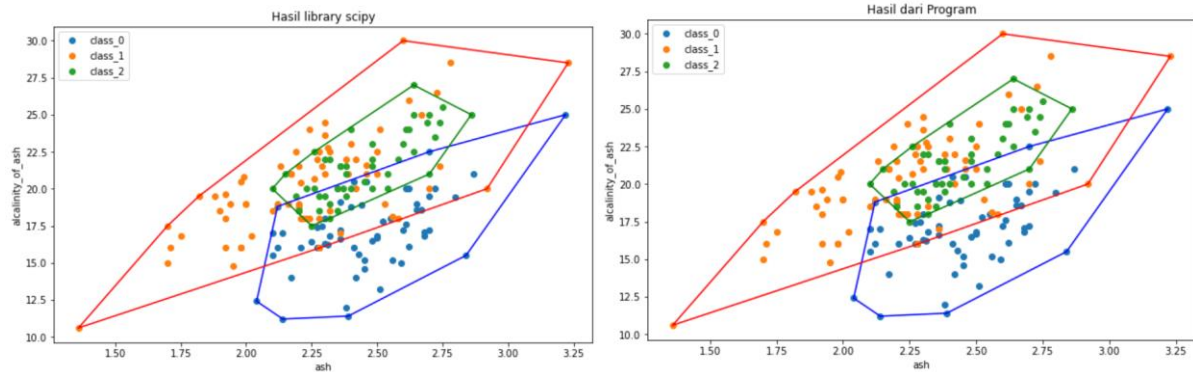
3.3 Alcohol vs Malic_acid (Wine)

Masukan dataset = 2, parameter 1 = 1, parameter 2 = 2



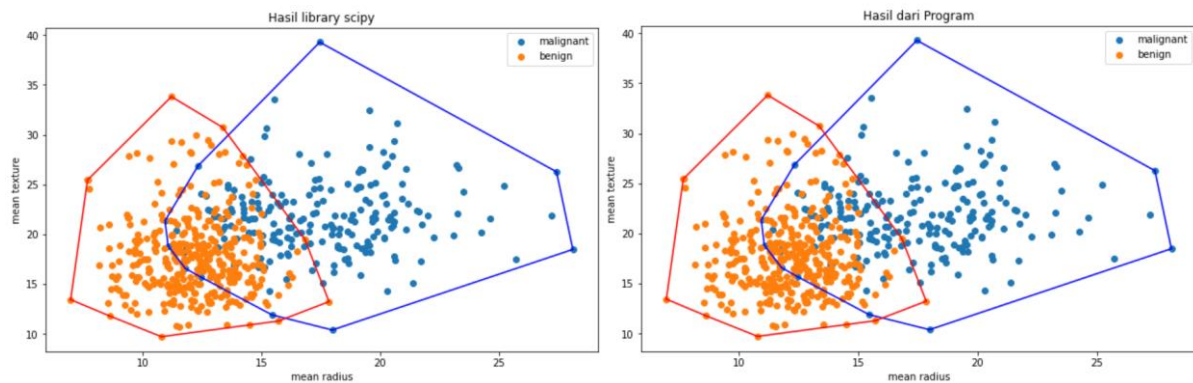
3.4 Ash vs Alcalinity_of_ash (Wine)

Masukan dataset = 2, parameter 1 = 3, parameter 2 = 4



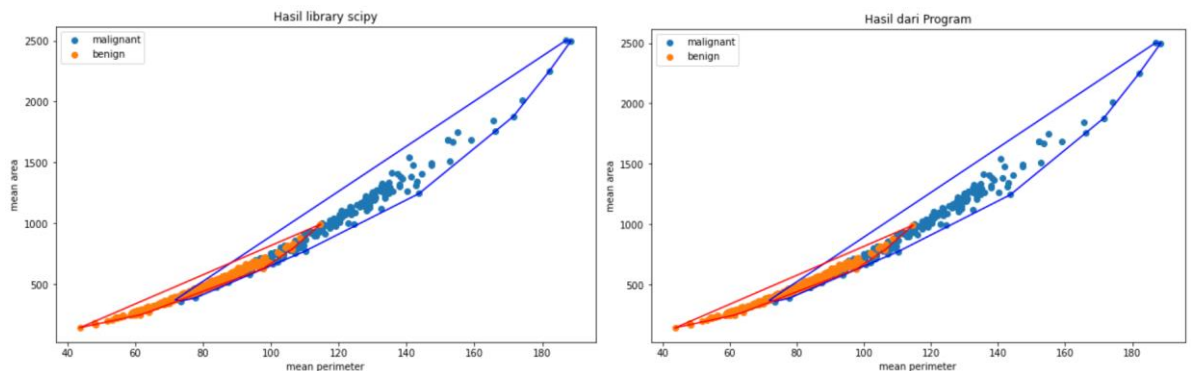
3.5 Mean radius vs Mean texture (Breast_cancer)

Masukan dataset = 3, parameter 1 = 1, parameter 2 = 2



3.6 Mean perimeter vs Mean area (Breast_cancer)

Masukan dataset = 1, parameter 1 = 3, parameter 2 = 4



LAMPIRAN

Link ke Repository Github

<https://github.com/Primanda28/tucil2-stima>

Tabel Penilaian

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	√	