**TUGAS KECIL 02**

**IMPLEMENTASI CONVEX HULL UNTUK VISUALISASI TES**

**LINEAR SEPARABILITY DATASET DENGAN**

**ALGORITMA DIVIDE AND CONQUER**

**IF2211 – STRATEGI ALGORITMA**

A picture containing text, clipart

Description automatically generated

**Disusun oleh:**

13520034 - Bryan Bernigen

**Institut Teknologi Bandung**

**Jl. Ganesha NO. 10, Bandung 40132**

**2022**

Daftar Isi

[Algoritma Convex Hull 3](#_Toc96644556)

[Algoritma Convex Hull Dengan Visualisasi 4](#_Toc96644557)

[Source Code 7](#_Toc96644558)

[myConvexHull.py 7](#_Toc96644559)

[main.py 10](#_Toc96644560)

[Hasil Pengujian 12](#_Toc96644561)

[1. Boston House Prices Dataset 12](#_Toc96644562)

[2. Iris Plants Dataset 12](#_Toc96644563)

[3. Diabetes Dataset 14](#_Toc96644564)

[4. Optical Recognition of Handwritten Digits Dataset 15](#_Toc96644565)

[5. Linnerund Dataset 16](#_Toc96644566)

[6. Wine Recognition Dataset 16](#_Toc96644567)

[7. Breast Cancer Wisconsin (Diagnostic) Dataset 18](#_Toc96644568)

[Lampiran 20](#_Toc96644569)

# Algoritma Convex Hull

Part 1: Jika belum memiliki 2 titik ekstrim

1. Urutkan data berdasarkan atribut ke-1 dan jika ada atribut ke-1 yang sama, urutkan berdasarkan atribut ke-2
2. Pilih 2 titik ekstrim (data ke-1 dan data terakhir)
3. Pisahkan data-data selain titik ekstrim menjadi 2 bagian yakni kiri dan kanan. Pembagian Dapat dilakukan dengan menggunakan determinan. Data dengan determinan positif berada di bagian kanan, data dengan determinan negatif di bagian kiri, dan hapus data dengan determinan 0
4. Lakukan Convex Hull Part 2 untuk bagian kiri dan kanan secara terpisah

Part 2: Jika sudah memiliki 2 titik ekstrim ex1 dan ex2

1. Jika array data kosong, maka diperoleh pasangan titik [ex1,ex2] yang akan menghasilkan convex hull
2. Jika array data tidak kosong, Cari titik extrim ex3 dari data-data yang ada dengan ex1 dan ex2. Dapat dicari dengan menggunakan jarak titik ke garis yang dibentuk oleh ex1 dan ex2
3. Hapus Titik-titik yang berada di dalam segitiga yang dibentuk oleh ex1, ex2 dan ex3. Dapat dicari dengan menggunakan luas segitiga karena A123 = A12x + A1x3 + Ax23 jika x di dalam segitiga A123
4. Pisahkan titik-titik sisanya menjadi bagian kiri dan kanan.
5. Lakukan Part 2 untuk data bagian kiri dengan e1,e3 sebagai titik extrim dan data bagian kanan dengan e3,e2 sebagai titik extrimnya.

# Algoritma Convex Hull Dengan Visualisasi

Part 1: Jika belum memiliki 2 titik ekstrim

1. Urutkan data berdasarkan atribut ke-1 dan jika ada atribut ke-1 yang sama, urutkan berdasarkan atribut ke-2

Sebelum

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| X | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| y | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 |

Sesudah:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Indeks | 1 | 4 | 7 | 2 | 5 | 8 | 3 | 6 | 9 |
| X | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 |
| y | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |

1. Pilih 2 titik ekstrim (data ke-1 dan data terakhir)

EX1 = [1,1,1]

EX2 = [9,3,3]

1. Pisahkan data-data selain titik ekstrim menjadi 2 bagian yakni kiri dan kanan. Pembagian Dapat dilakukan dengan menggunakan determinan. Data dengan determinan positif berada di bagian kanan, data dengan determinan negatif di bagian kiri, dan hapus data dengan determinan 0

|  |  |  |
| --- | --- | --- |
| 7 | 8 | EX2 |
| 4 | 5 | 6 |
| EX1 | 2 | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Kiri | | |  | Kanan | | |
| Indeks | X | Y |  | Indeks | X | Y |
| 7 | 1 | 3 |  | 2 | 2 | 1 |
| 4 | 1 | 2 |  | 3 | 3 | 1 |
| 8 | 2 | 3 |  | 6 | 3 | 2 |

1. Lakukan Convex Hull Part 2 untuk bagian kiri dan kanan secara terpisah

Part 2: Jika sudah memiliki 2 titik ekstrim ex1 dan ex2

\*contoh: EX1=[1,1,1] , EX2=[9,3,3], dan data = [[7,1,3],[4,1,2],[8,2,3]]

1. Jika array data kosong, maka diperoleh pasangan titik [ex1,ex2] yang akan menghasilkan convex hull

Misal: EX1=[1,1,1] , EX2=[7,1,3], dan data = [[]] 🡪 Hasil = [[1,7]]

1. Jika array data tidak kosong, Cari titik extrim ex3 dari data-data yang ada dengan ex1 dan ex2. Dapat dicari dengan menggunakan jarak titik ke garis yang dibentuk oleh ex1 dan ex2

|  |  |  |
| --- | --- | --- |
| 7 | 8 | EX2 |
| 4 |  |  |
| EX1 |  |  |

Karena jarak dari titik 7 ke garis paling jauh 🡪 EX3 = [7,1,3]

1. Hapus Titik-titik yang berada di dalam segitiga yang dibentuk oleh ex1, ex2 dan ex3. Dapat dicari dengan menggunakan luas segitiga karena A123 = A12x + A1x3 + Ax23 jika x di dalam segitiga A123

|  |  |  |
| --- | --- | --- |
| EX3 | 8 | EX2 |
| 4 |  |  |
| EX1 |  |  |

Karena titik 4 dan 8 berada dalam segitiga EX1,EX2,EX3 🡪 hapus dari data

1. Pisahkan titik-titik sisanya menjadi bagian kiri dan kanan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | EX3 |  | EX2 |  |
|  |  |  |  |  |
|  | EX1 |  |  |  |
|  |  |  |  |  |

Partisi dapat dilakukan seperti garis hijau maupun ungu.

Misal terdapat 2 titik seperti pada gambar, maka titik merah akan masuk ke bagian kiri dan titik biru akan masuk ke bagian kanan

1. Lakukan Part 2 untuk data bagian kiri dengan e1,e3 sebagai titik extrim dan data bagian kanan dengan e3,e2 sebagai titik extrimnya.

Ulangi Convex Hull part2 dengan data sebagai berikut

|  |  |  |
| --- | --- | --- |
|  | EX3 |  |
|  |  |  |
|  | EX1 |  |

# Source Code

## myConvexHull.py

berisi seluruh fungsi yang diperlukan untuk menghasilkan Convex Hull

import numpy as np

from numpy.linalg import norm

#Menambahkan Nomor titik pada array of titik sekaligus mengurutkan data

def addNo(bucket):

    temp = []

    for i in range(len(bucket)):

        temp.append([i, bucket[i][0], bucket[i][1]])

    # Mengurutkan Data sesuai x dan y

    temp = sorted(temp, key=lambda x: (x[1], x[2]))

    return temp

# mengecek jika test didalam segitiga t0,t1,t2 dengan menggunakan area

#jika test bernilai true jika test berada dalam segitiga t1 t2 t3

def dalemSegitiga(t1, t2, t3, test):

    #menghitung luas segitiga

    def area(x1, y1, x2, y2, x3, y3):

        return abs((x1 \* (y2 - y3) + x2 \* (y3 - y1)

                + x3 \* (y1 - y2)) / 2.0)

    #menghitung luas segitiga dari kombinasi 4 titik

    A123 = area(t1[1], t1[2], t2[1], t2[2], t3[1], t3[2])

    Atest23 = area(test[1], test[2], t2[1], t2[2], t3[1], t3[2])

    A1test3 = area(t1[1], t1[2], test[1], test[2], t3[1], t3[2])

    A12test = area(t1[1], t1[2], t2[1], t2[2], test[1], test[2])

    if(A123 - (Atest23 + A1test3 + A12test) > -10e-6 and A123 - (Atest23 + A1test3 + A12test) < 10e-6):

        return True

    else:

        return False

# mengecek jika titik test dikiri atau dikanan garis t1 t2

# bernilai true jika test berada di kanan garis t1 t2

def kiriOrKanan(t1, t2, test):

    result = t1[1]\*t2[2] + test[1]\*t1[2] + t2[1] \* \

        test[2] - test[1]\*t2[2]-t2[1]\*t1[2] - t1[1]\*test[2]

    if(result == 0):

        return -1

    if(result > 0):

        return 1

    else:

        return 0

# menghitung jarak titik test ke sebuah garis t1 t2

def jarakTitikKeGaris(t1, t2, test):

    p1 = [t1[1], t1[2]]

    p2 = [t2[1], t2[2]]

    p3 = [test[1], test[2]]

    p1 = np.asarray(p1)

    p2 = np.asarray(p2)

    p3 = np.asarray(p3)

    d = norm(np.cross(p2-p1, p1-p3))/norm(p2-p1)

    return d

# convexhull jika masih belum memilih 2 titik maximal

def MyConvexHull(bucket):

    bucket = addNo(bucket)

    t1 = bucket[0]

    t2 = bucket[len(bucket)-1]

    if(bucket == []):

        return[[t1[0], t2[0]]]

    else:

        kiri = []

        kanan = []

        for i in range(len(bucket)):

            if(kiriOrKanan(t1, t2, bucket[i]) == 1):

                kanan += ([bucket[i]])

            if(kiriOrKanan(t1, t2, bucket[i]) == 0):

                kiri += ([bucket[i]])

        a = smallerConvexHull(kanan, t1, t2)

        b = smallerConvexHull(kiri, t1, t2)

        a += b

        return(a)

# convexhull jika sudah ada 2 titik maximal

def smallerConvexHull(bucket, t1, t2):

    if(bucket == []):

        return[[t1[0], t2[0]]]

    else:

        titikmax = [0, 0, 0]

        jarakmax = 0

        for i in range(len(bucket)):

            d = jarakTitikKeGaris(t1, t2, bucket[i])

            if(d > jarakmax):

                titikmax = bucket[i]

                jarakmax = d

        luar = []

        for i in range(len(bucket)):

            if(not dalemSegitiga(t1, t2, titikmax, bucket[i])):

                luar.append(bucket[i])

        kanan = []

        kiri = []

        for i in range(len(luar)):

            if(luar[i][1] < titikmax[1]):

                kiri.append(luar[i])

            else:

                kanan.append(luar[i])

        a = smallerConvexHull(kiri, t1, titikmax)

        b = smallerConvexHull(kanan, titikmax, t2)

        a += (b)

        return(a)

## main.py

program utama yang akan dijalankan. Main akan mempersiapkan data untuk dicari convex hullnya. Main akan memanggil fungsi myConvexHull di modul myConvexHull.py

from audioop import mul

from json import load

from scipy.spatial import ConvexHull

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import datasets, multiclass

from myConvexHull import MyConvexHull

#UBAH DATA YANG INGIN DI PLOT DI SINI=======================================

data = datasets.load\_boston()

#UBAH DATA YANG INGIN DI PLOT DI SINI=======================================

# create a DataFrame

df = pd.DataFrame(data.data, columns=data.feature\_names)

df['Target'] = pd.DataFrame(data.target)

# visualisasi hasil ConvexHull

#masukkan index kolom data yang ingin di plot

#UBAH YANG INGIN DI PLOT DI BAGIAN SINI=====================================

#index parameter yang akan di plot di sumbu\_x dan sumbu\_y

sumbu\_x= 5

sumbu\_y = 9

judul\_plot = 'Average Number of Rooms vs Tax Rate per Year'

colors = ['b', 'r', 'g']

#UBAH YANG INGIN DI PLOT DI BAGIAN SINI=====================================

plt.figure(figsize=(10, 6))

plt.title(judul\_plot)

plt.xlabel(data.feature\_names[sumbu\_x])

plt.ylabel(data.feature\_names[sumbu\_y])

#CEK TABEL DATA YANG AKAN DI PLOT===========================================

multiple\_class=True

multiple\_taget=True

#cek banyaknya kelas

try:

    data.target\_names

except:

    multiple\_class=False

#cek tipe target

try:

    a=df[df['Target']==0]

    a[0]

except:

    multiple\_taget=False

loop=1

if (multiple\_class):

    loop = len(data.target\_names)

#CEK TABEL DATA YANG AKAN DI PLOT===========================================

for i in range(loop):

    if(multiple\_taget):

        bucket = df[df['Target']==i]

    else:

        bucket = df[df['Target']>0]

    bucket = bucket.iloc[:, [sumbu\_x, sumbu\_y]].values

    #CONVEX HULL USING MY OWN ALGORITHM===================================================

    hull = MyConvexHull(bucket)

    if(multiple\_class):

        plt.scatter(bucket[:, 0], bucket[:, 1],label=data.target\_names[i])

    else:

        plt.scatter(bucket[:, 0], bucket[:, 1],label=judul\_plot)

    for simplex in hull:

        plt.plot(bucket[simplex, 0], bucket[simplex, 1], colors[i])

    #CONVEX HULL USING MY OWN ALGORITHM===================================================

plt.legend()

plt.show()

# Hasil Pengujian

## Boston House Prices Dataset

1. Crime per Capita vs Lower Status % of the Population

Chart, scatter chart

Description automatically generated

1. Average Number of Rooms vs Tax Rate per Year

Chart, scatter chart

Description automatically generated

## Iris Plants Dataset

1. Petal Width vs Petal Legth

Chart, scatter chart

Description automatically generated

1. Sepal Width vs Sepal Lenght

Chart

Description automatically generated

Catatan: Terdapat titik berwarna hijau yang masuk ke garis oranye. Hasil tersebut juga terjadi karena titik tersebut masuk ke kelas oranye dan hijau namun hijau di print setelah oranye sehingga warna menjadi hijau

Chart

Description automatically generated

## Diabetes Dataset

1. Age vs BMI

Chart, scatter chart

Description automatically generated

1. Age vs Average Blood Pressure

Chart, scatter chart

Description automatically generated

1. BMI vs Average Blood Pressure

Chart, scatter chart

Description automatically generated

## Optical Recognition of Handwritten Digits Dataset

Text

Description automatically generated

Error karena deskripsi dataset kurang jelas

## Linnerund Dataset



Error karena atribut dataset berupa array dengan 3 atribut

Graphical user interface, text, application, email

Description automatically generated

## Wine Recognition Dataset

1. Ash vs Alcalanity of Ash

Chart

Description automatically generated

1. Total phenols vs Nonflavanoid phenols

Chart, radar chart

Description automatically generated

Chart, scatter chart

Description automatically generated

1. Color Intensity vs Hue

Chart

Description automatically generated

## Breast Cancer Wisconsin (Diagnostic) Dataset

1. Mean Radius vs Mean Texture

Chart, scatter chart

Description automatically generated

1. Mean Smoothness vs Mean Compactness

Chart, radar chart, scatter chart

Description automatically generated

1. Mean Concavity vs Mean Concave Points

Chart, scatter chart

Description automatically generated

# Lampiran

Table

Description automatically generated

√

√

√

√