LAPORAN TUGAS KECIL 2 IF2211 STRATEGI ALGORITMA

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

PUTRI NURHALIZA

13520066

TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

SEMESTER II 2021/202

Daftar Isi

A. Algoritma *Divide and Conquer* pada Convex Hull

Himpunan titik pada bidang planar disebut convex jika untuk sembarang dua titik pada bidang tersebut (misal p dan q), seluruh segmen garis yang berakhir di p dan q berada pada himpunan tersebut. Algoritma *divide and conquer* digunakan pada pustaka myConvexHull untuk menemukan titik-titik terluar yang membentuk convex hull.

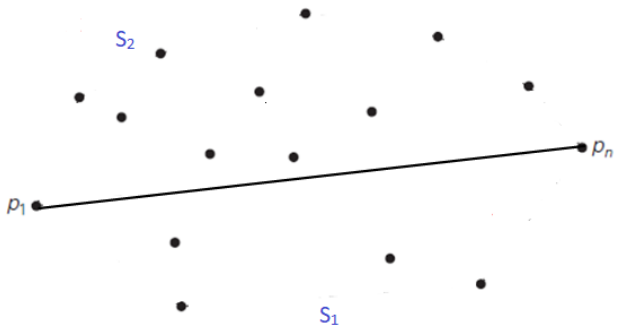
Sebelum dilakukan *divide*, n titik pada S di bidang kartesian dua dimensi diurutkan berdasarkan nilai absis yang menaik, dan ordinat yang menaik jika ada nilai absis yang sama. Kemudian tentukan p_1 dan p_n yang merupakan titik ekstrim yang akan membentuk convex hull tersebut. Setelah itu, dilakukan tahap *divide* sebagai berikut.

1. S dibagi menjadi S_1 dan S_2 oleh garis yang menghubungkan p_1 dan p_n (p_1p_n). Jika p_1 ke p_n dianggap memiliki orientasi kedepan, maka S_1 berada di sebelah kanan, dan S_2 di sebelah kiri. Pemilahan titik-titik ini ditentukan berdasarkan determinan. Misal p_1 (x_1, y_1), p_n (x_2, y_2), dan p_{lain} (x_3, y_3), maka determinannya adalah sebagai berikut.

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = x_1y_2 + x_3y_1 + x_2y_3 - x_3y_2 - x_2y_1 - x_1y_3$$

Gambar 1.1. Mencari determinan untuk menentukan arah

Jika hasil determinan positif, artinya titik p_{lain} (x_3, y_3) berada di sebelah kiri garis p_1p_n, dan sebaliknya. Jika hasil determinan sama dengan 0, artinya titik tersebut berada pada garis p_1p_n sehingga tidak akan dimasukkan ke S_1 maupun S_2 untuk perhitungan lebih lanjut.

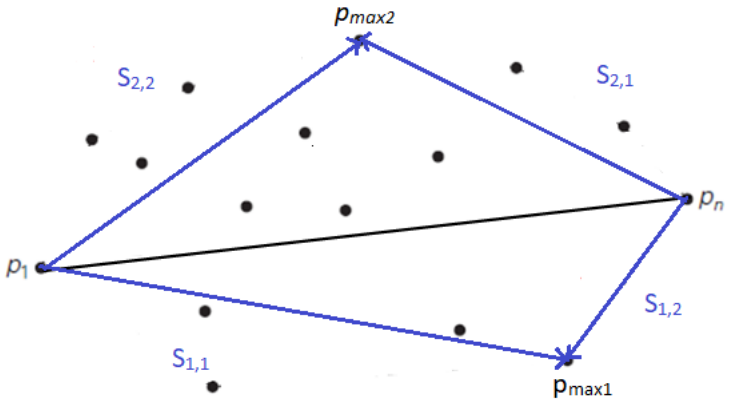


Gambar 1.2. Ilustrasi membagi titik-titik menjadi 2 bagian

2. Selanjutnya, akan dicari convex hull pada S_1 maupun S_2 dengan metode rekursif yang kemudian akan membentuk convex hull S secara keseluruhan.
3. Pada S_1,
   a. Jika S_1 kosong maka tidak ada titik lain yang menjadi pembentuk convex hull pada bagian tersebut.
   b. Jika S_1 tidak kosong, pilih sebuah titik (misal p_{max1}) yang berjarak maksimal dari garis p_1p_n. p_{max1} akan terhitung sebagai salah satu titik pembentuk convex hull.
   c. Segitiga $p_1p_{max1}p_n$ akan membagi S_1 menjadi 3 bagian. Titik-titik yang berada di dalam segitiga maupun di garis p_1p_{max1} atau garis p_np_{max1} dapat diabaikan.

d. Selanjutnya, tentukan titik titik-titik yang berada di sebelah kanan ($S_{1,1}$) dengan melihat determinannya terhadap $p_1 p_{max1}$, serta titik-titik yang berada di sebelah kiri (misal $S_{1,2}$) dengan melihat determinannya terhadap $p_n p_{max1}$.
e. Lakukan poin 3 secara rekursif hingga bagian kanan dan kiri kosong.

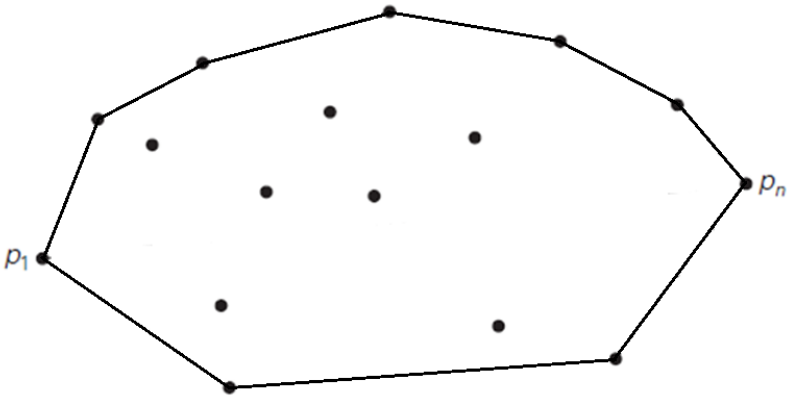
4. Begitupun pada S_2, lakukan hal yang sama seperti poin 3. Namun, ada sedikit perbedaan pada poin d untuk menjaga konsistensi orientasi. Tentukan titik titik-titik yang berada di sebelah kanan ($S_{2,1}$) dengan melihat determinannya terhadap $p_n p_{max2}$, serta titik-titik yang berada di sebelah kiri (misal $S_{2,2}$) dengan melihat determinannya terhadap $p_1 p_{max2}$, serta. Hal ini juga dilakukan secara rekursif hingga bagian kanan dan kiri kosong.



Gambar 1.3. Ilustrasi menentukan p_{max} tiap bagian

Pada tahap *combine*, titik p_1, p_n, serta p_{max} dari hasil rekursi digabungkan sebagai convex hull. Tahap penggabungan ini sebenarnya tidak memerhatikan urutan. Namun, untuk mempermudah plot pada visualisasi, titik-titik tersebut dapat diurutkan searah jarum jam maupun berlawanan arah jarum jam. Sebagai contoh, untuk arah berlawanan jarum jam dapat dilakukan dengan cara berikut,

- Pada S_1 maupun S_2, urutannya adalah $kanan - p_{max} - kiri$ (kanan dan kiri berlaku rekursif)
- Untuk menggabungkan keseluruhan, urutannya adalah $p_1 - S_1 - p_n - S_2$



Gambar 1.4. Ilustrasi hasil akhir convex hull

B. *Source Code* Program (Bahasa Pemograman Python)

Source code program dikelompokkan menjadi tiga.

1. MyConvexHull.py
   Pustaka yang digunakan untuk menentukan mendapatkan convex hull.

```
import numpy as np

def orientation(p1, p2, p3):
    """
        prekondisi: p1, p2, dan p3 merupakan array yang terdefenisi, merepresentasikan titik
        proses: Menentukan posisi titik p3 terhadap garis p1p2 berdasarkan determinan
    """
    det = p1[0]*p2[1] + p2[0]*p3[1] + p3[0]*p1[1] - p3[0]*p2[1] - p2[0]*p1[1] - p1[0]*p3[1]
    if (np.allclose(det, 0)): det = 0.0

    if (det > 0): return 1          # left
    elif (det < 0): return -1       # right
    return 0                        # in line

def pointMax(sub, p1, p2):
    """
        prekondisi: array sub tidak kosong
                    p1 dan p2 merupakan array yang terdefenisi, merepresentasikan titik-titik
        proses: menentukan titik pada array sub yang memiliki jarak terjauh terhadap garis
p1p2
                jika ada beberapa titik dengan jarak terjauh, akan terpilih yang pertama
    """
    max = 0.0
    pMax = np.array([])
    for p in sub:
        d = np.linalg.norm(np.cross(p2-p1, p1-p))/np.linalg.norm(p2-p1)
        if (d > max):
            max = d
            pMax = p

    return pMax

def addHull(sub, pBegin, pEnd):
    """
        prekondisi: array sub terdefenisi, boleh kosong
                    pBegin dan pEnd merupakan array yang terdefenisi, merepresentasikan titik
                    dan merupakan convex hull yang sudah diketahui di sisi tersebut
        proses: menentukan convex hull pada suatu bagian, dilakukan secara rekursif
    """
    if len(sub) == 0:
        return np.empty((0,2), float)
    else:
        pMax = pointMax(sub, pBegin, pEnd)
```

```
        result = right = left = np.empty((0,2), float)

        for point in sub:
            if orientation(pBegin, pMax, point) == 1: left = np.append(left, [point], axis=0)
            elif orientation(pEnd, pMax, point) == -1: right = np.append(right, [point],
axis=0)

        result = np.append(result, addHull(right, pMax, pEnd), axis=0)
        result = np.append(result, [pMax], axis=0)
        result = np.append(result, addHull(left, pBegin, pMax), axis=0)

        return result

def myConvexHull(points):
    """
        prekondisi: array points terdefenisi, minimal terdapat 2 titik
        proses: melakukan divide and conquer untuk menentukan convex hull keseluruhan
    """
    points = points[np.lexsort((points[:, 1], points[:, 0]))]
    p1 = points[0]
    pn = points[-1]
    result = right = left = np.empty((0,2), float)

    for point in points:
        if orientation(p1, pn, point) == 1: left = np.append(left, [point], axis=0)
        elif orientation(p1, pn, point) == -1: right = np.append(right, [point], axis=0)

    result = np.append(result, [p1], axis=0)
    result = np.append(result, addHull(right, pn, p1), axis=0)
    result = np.append(result, [pn], axis=0)
    result = np.append(result, addHull(left, p1, pn), axis=0)

    return result
```

2. Visualization.py
   Menampilkan visualisasi convex hull yang dihasilkan dari suatu dataset dan atribut yang dipilih *user*. Visualisasi akan ditampilkan dengan *pop up* matplotlib. User perlu menutup *preview* tersebut untuk melanjutkan program.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from MyConvexHull import myConvexHull

def visualization(data, x, y, dsName):
    """
        prekondisi: data terdefinisi untuk suatu dataset dsName, x dan y indeks atribut data
yang valid
        proses: membentuk dataframe
```

```
                menentukan convex hull dari data dengan myConvexhull()
                menampilkan visualisasi
                menyimpan visualisasi jika diinginkan user
    """
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)

    plt.figure(figsize = (10, 6))
    colors = ['b','orange','g','r', 'purple', 'brown', 'pink', 'gray', 'y', 'cyan']
    title = f"{data.feature_names[x].upper()} vs {data.feature_names[y].upper()} ({dsName})"
    plt.title(title)
    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
        plt.scatter(bucket[:, 0], bucket[:, 1], label=data.target_names[i])
        hull = myConvexHull(bucket)
        # print(hull)
        plt.plot(np.append(hull[:, 0], hull[0, 0]), np.append(hull[:, 1], hull[0, 1]),
colors[i])
    plt.legend()

    save = input("\nSave image (y/n)? ")
    if (save == 'y'):
        plt.savefig('../test/' + title + '.png')
    plt.show()
```

3. main.py
   Program utama yang berisi alur program serta *interaksi* dengan user. Termasuk sebuah fungsi untuk *load* dataset yang diinginkan. Program akan terus berjalan jika *user* masih ingin mencoba tes yang lain.

```
from Visualization import *
from sklearn import datasets

def loadData(opt):
    """
        prekondisi: opt memiliki nilai yang valid untuk refer ke dataset (1-4)
        proses: meload data yang sesuai
                menginput pilihan attribute dari user
                memanggil fungsi visualisasi jika valid
    """
    dsNames = ["IRIS", "WINE", "BREAST_CANCER", "DIGITS"]
    if (opt == 1): data = datasets.load_iris()
    elif (opt == 2): data = datasets.load_wine()
    elif (opt == 3): data = datasets.load_breast_cancer()
```

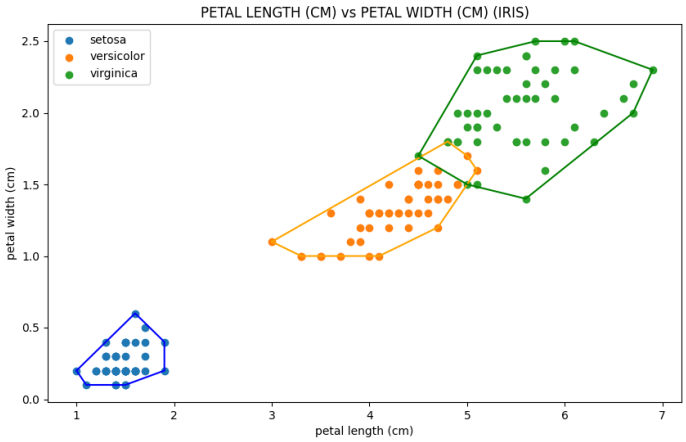
```
    elif (opt == 4): data = datasets.load_digits()

    print(f"\nAvailable attributes: ")
    for i in range(len(data.feature_names)):
        print(f"{i+1}. {data.feature_names[i]}")
    print(f"\nChoose attribute (x and y): ")
    x = int(input("x: ")) - 1
    y = int(input("y: ")) - 1

    maxCol = len(data.feature_names) - 1
    if (x < 0 or x > maxCol or y < 0 or y > maxCol):
        print("The choosen attribute is invalid")
    else:
        visualization(data, x, y, dsNames[opt-1])



# --------------------------- PROGRAM UTAMA -------------------------------

print("==================LINEAR SEPARABILITY TEST==================")

isRun = True
while (isRun):
    print("\n1. Iris")
    print("2. Wine")
    print("3. Breast_cancer")
    print("4. Digits")
    opt = int(input("Choose dataset: "))
    if (opt > 0 and opt <= 4):
        loadData(opt)
    else:
        print("Dataset not available")

    run = input("\nTry another test (y/n)? ")
    isRun = True if run == 'y' else False
```

C. Input dan Output Hasil Eksekusi

Program akan meminta *user* untuk menginput pilihan dataset yang ingin diuji beserta atributnya. Hasil visualisasi akan disimpan di dalam folder */test* jika diinginkan.

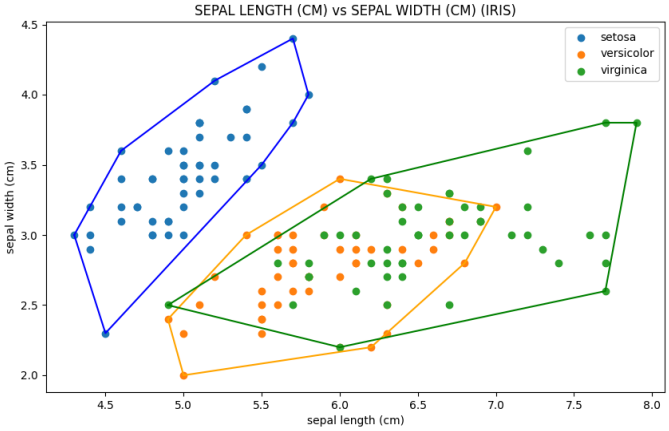
1. Petal length vs Petal Width (IRIS)

```
1. Iris
2. Wine
3. Breast_cancer
4. Digits
Choose dataset: 1

Available attributes:
1. sepal length (cm)
2. sepal width (cm)
3. petal length (cm)
4. petal width (cm)

Choose attribute (x and y):
x: 3
y: 4
```



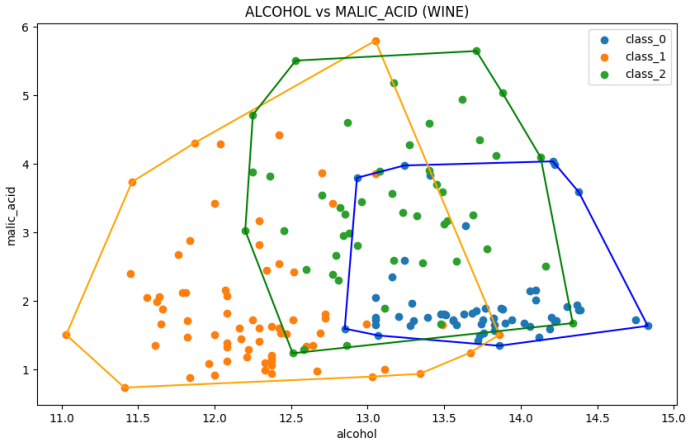
2. Sepal length vs Sepal Width (IRIS)

```
1. Iris
2. Wine
3. Breast_cancer
4. Digits
Choose dataset: 1

Available attributes:
1. sepal length (cm)
2. sepal width (cm)
3. petal length (cm)
4. petal width (cm)

Choose attribute (x and y):
x: 1
y: 2
```



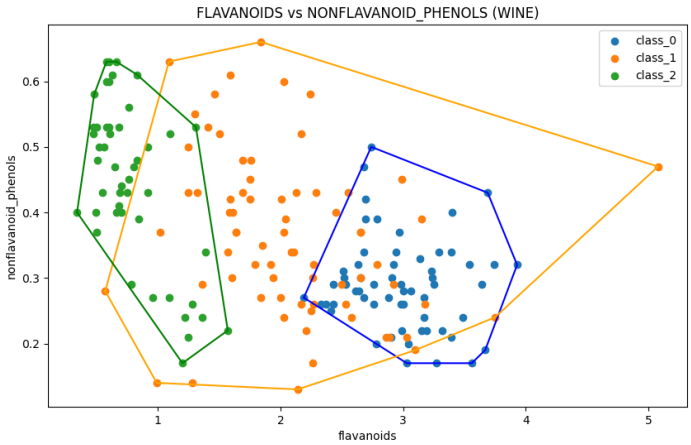
3. Alcohol vs Malic_acid (WINE)

```
1. Iris
2. Wine
3. Breast_cancer
4. Digits
Choose dataset: 2

Available attributes:
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

Choose attribute (x and y):
x: 1
y: 2
```



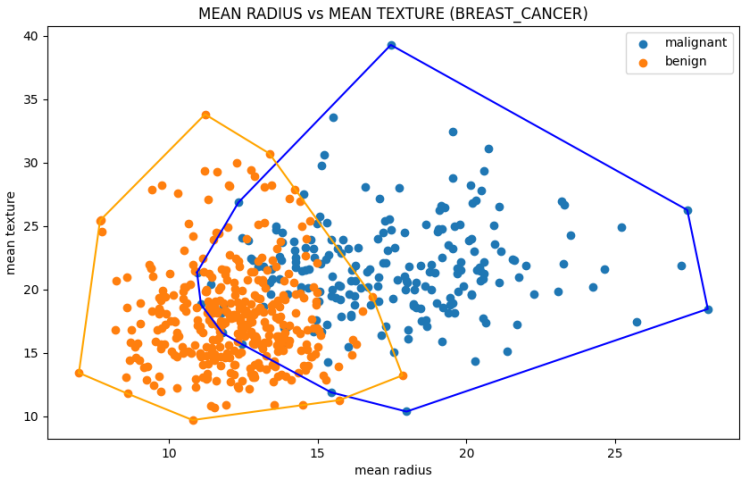
4. Flavanoids vs Nonflavanoid_phenols (WINE)

```
1. Iris
2. Wine
3. Breast_cancer
4. Digits
Choose dataset: 2

Available attributes:
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

Choose attribute (x and y):
x: 7
y: 8
```



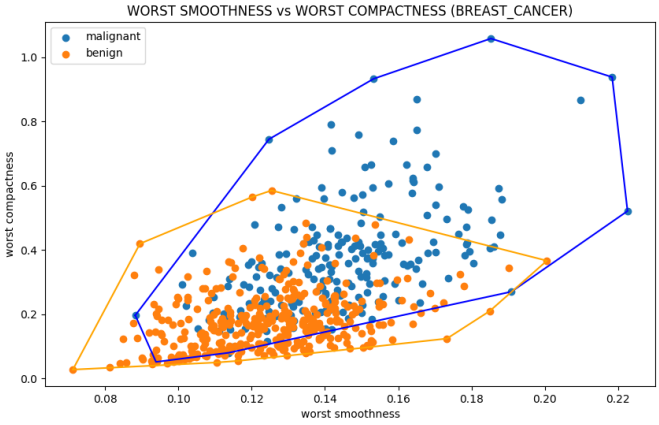
5. Mean Radius vs Mean Texture (BREAST_CANCER)

```
1. Iris
2. Wine
3. Breast_cancer
4. Digits
Choose dataset: 3

Available attributes:
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

Choose attribute (x and y):
x: 1
y: 2
```



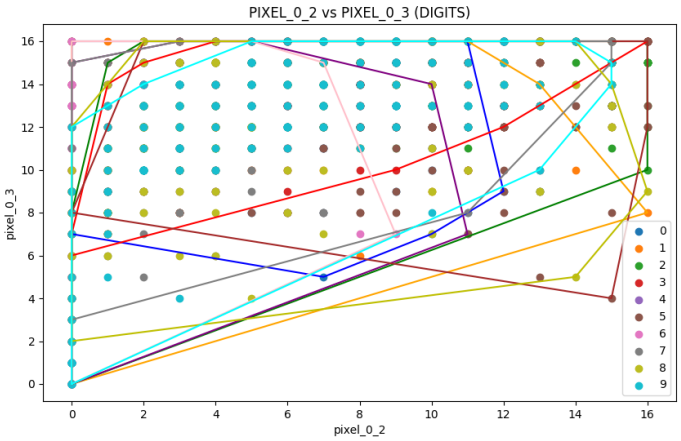
6. Worst Smoothness vs Worst Compactness (BREAST_CANCER)

```
Choose attribute (x and y):
x: 25
y: 26
```

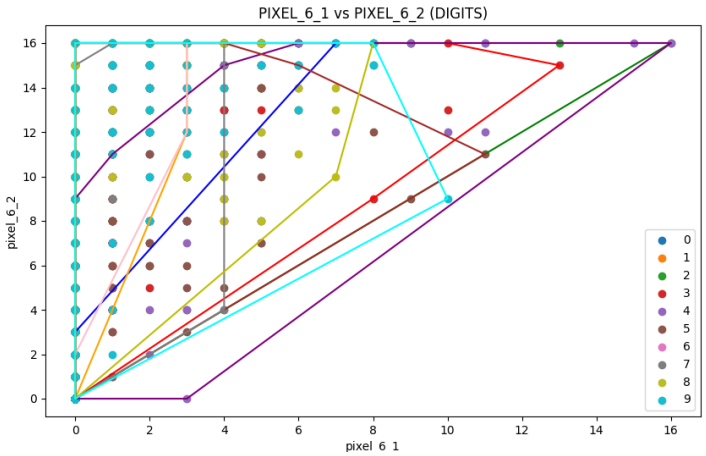


7. Pixel_0_2 vs Pixel_0_3 (DIGITS)

```
1. Iris
2. Wine
3. Breast_cancer
4. Digits
Choose dataset: 4

Available attributes:
1. pixel_0_0
2. pixel_0_1
3. pixel_0_2
4. pixel_0_3
5. pixel_0_4
6. pixel_0_5
7. pixel_0_6
8. pixel_0_7
9. pixel_1_0
10. pixel_1_1
11. pixel_1_2
12. pixel_1_3
13. pixel_1_4
14. pixel_1_5
15. pixel_1_6
16. pixel_1_7
17. pixel_2_0
18. pixel_2_1
19. pixel_2_2
20. pixel_2_3
21. pixel_2_4
22. pixel_2_5
23. pixel_2_6
24. pixel_2_7
25. pixel_3_0
26. pixel_3_1
27. pixel_3_2
28. pixel_3_3
29. pixel_3_4
30. pixel_3_5
31. pixel_3_6
32. pixel_3_7
33. pixel_4_0
34. pixel_4_1
35. pixel_4_2
36. pixel_4_3
37. pixel_4_4
38. pixel_4_5
39. pixel_4_6
40. pixel_4_7
41. pixel_5_0
42. pixel_5_1
43. pixel_5_2
44. pixel_5_3
45. pixel_5_4
46. pixel_5_5
47. pixel_5_6
48. pixel_5_7
49. pixel_6_0
50. pixel_6_1
51. pixel_6_2
52. pixel_6_3
53. pixel_6_4
54. pixel_6_5
55. pixel_6_6
56. pixel_6_7
57. pixel_7_0
58. pixel_7_1
59. pixel_7_2
60. pixel_7_3
61. pixel_7_4
62. pixel_7_5
63. pixel_7_6
64. pixel_7_7

Choose attribute (x and y):
x: 3
y: 4
```



8. Pixel_6_1 vs Pixel_6_2 (DIGITS)




```
Choose attribute (x and y):
x: 50
y: 51
```

D. *Repository* Program (Github)

Untuk menjalankan program, *source code* bisa didapatkan di

https://github.com/Putriliza/Tucil2Stima-ConvexHull

E. Rangkuman Keberjalanan Program

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.	✓	-