

Laporan Tugas Kecil 2

Algoritma Divide and Conquer

Dibuat untuk memenuhi tugas pada mata kuliah Strategi Algoritma IF2211 Semester 2  
Tahun 2021/2022



Oleh

Willy Wilsen – 13520160

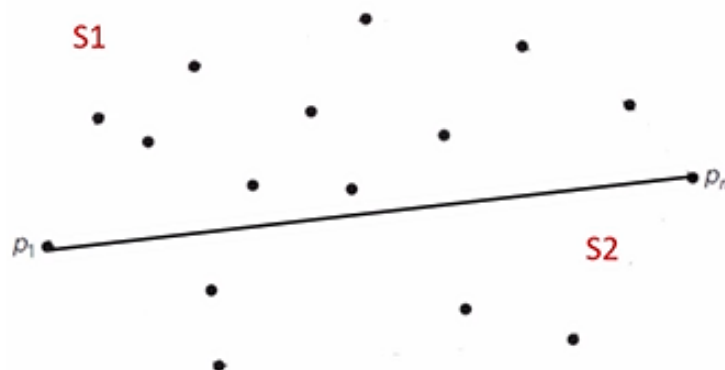
## 1. Algoritma Divide and Conquer pada Convex Hull

Misalkan diberikan kumpulan titik  $S$  seperti gambar berikut.



Algoritma Divide and Conquer:

Mula-mula, pilih suatu titik  $p_1 (x_1, y_1)$  dan  $p_n (x_n, y_n)$  yang merupakan dua titik ekstrim yang akan membentuk convex hull untuk kumpulan titik tersebut. Titik ekstrim yang dimaksud adalah titik dengan nilai absis terendah dan tertinggi. Jika nilai absis sama, maka dipilih berdasarkan nilai ordinat yang menaik. Kemudian garis yang menghubungkan  $p_1$  dan  $p_n$  membagi kumpulan titik  $S$  menjadi dua bagian yaitu  $S_1$  (kumpulan titik di sebelah kiri atau atas garis  $p_1 p_n$ ) dan  $S_2$  (kumpulan titik di sebelah kanan atau bawah garis  $p_1 p_n$ ) seperti gambar berikut.



Untuk memeriksa apakah sebuah titik berada di sebelah kiri suatu garis yang dibentuk dua titik dapat menggunakan persamaan determinan:

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

$$= x_1 y_2 + x_2 y_3 + x_3 y_1 - x_3 y_2 - x_2 y_1 - x_1 y_3$$

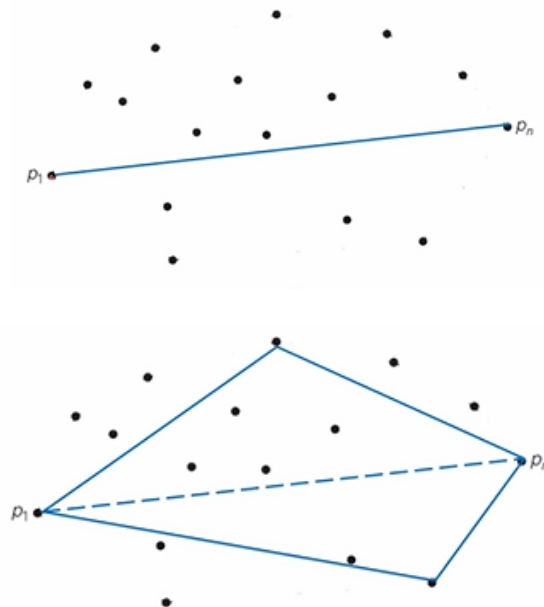
$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

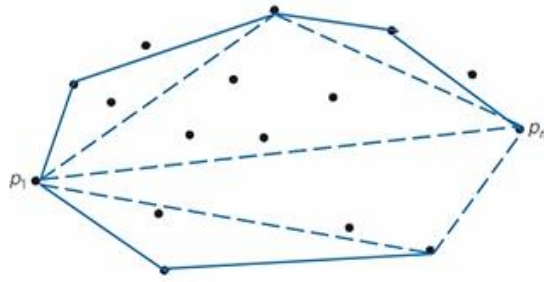
Titik  $(x_3, y_3)$  berada di sebelah kiri dari garis yang dibentuk  $(x_1, y_1)$  dan  $(x_2, y_2)$  jika hasil determinan positif. Semua titik yang berada pada garis  $p_1p_n$  tidak mungkin membentuk convex hull sehingga bisa diabaikan dari pemeriksaan. Lalu, kumpulan titik pada  $S_1$  dapat membentuk convex hull bagian atas dan kumpulan titik pada  $S_2$  dapat membentuk convex hull bagian bawah seperti gambar berikut.

Untuk sebuah bagian (misal  $S_1$ ) terdapat dua kemungkinan:

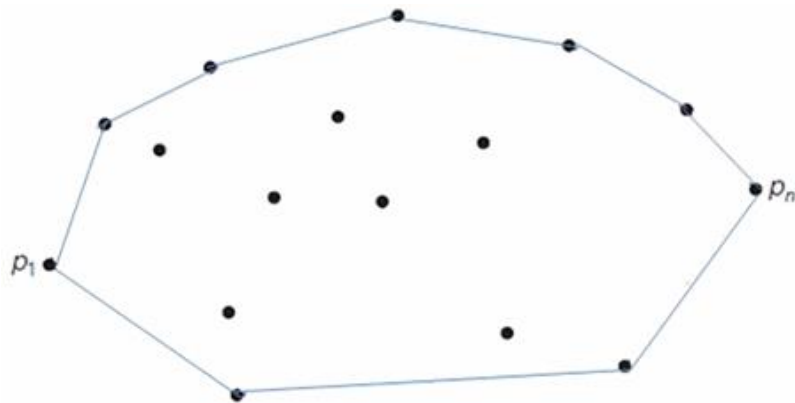
- Jika tidak ada titik lain selain  $p_1$  dan  $p_n$  di  $S_1$ , maka titik  $p_1$  dan  $p_n$  menjadi pembentuk convex hull bagian  $S_1$  (**Basis dari algoritma Divide and Conquer**).
- Jika  $S_1$  tidak kosong, pilih suatu titik yang memiliki jarak terjauh dari garis  $p_1p_n$  (misal  $p_{\max}$ ). Kemudian, semua titik yang berada di dalam daerah segitiga  $p_{\max}p_1p_n$  diabaikan untuk pemeriksaan lebih lanjut (karena tidak mungkin membentuk convex hull) dan semua titik yang berada di luar daerah segitiga  $p_{\max}p_1p_n$  dilakukan pemeriksaan lebih lanjut untuk menemukan titik pembentuk convex hullnya (**Rekursif dari algoritma Divide and Conquer**).

Ilustrasi algoritma Divide and Conquer dapat dilihat pada gambar berikut.





Maka, hasil akhir dari algoritma Divide and Conquer pada Convex Hull untuk kumpulan titik S adalah sebagai berikut.



## 2. Kode program

Kode program dibuat menggunakan bahasa Python dengan nama file yaitu "myConvexHull.py". Berikut fungsi, kelas, dan potongan code yang terdapat dalam file tersebut.

No	Nama Fungsi	Keterangan
1	extreme(M)	Menerima parameter sebuah matriks M yang berisi nilai absis dan ordinat setiap titik kemudian mengembalikan sebuah array 2 elemen yang berisi index dari titik ekstrim minimum dan maksimum
2	determinan(p1,p2,array)	Menerima parameter titik p1, p2 dan array (p3) kemudian mengembalikan nilai determinan titik array (p3) terhadap garis p1p2
3	leftPoints(p1,p2,M,left)	Menerima parameter titik p1, p2, matriks M, dan boolean left kemudian mengembalikan array yang

		berisi indeks dari titik-titik yang berada di sebelah kiri (atau atas) garis p1p2 jika left bernilai True dan array yang berisi indeks dari titik-titik yang berada di sebelah kanan (atau bawah) garis p1p2 jika left bernilai False
4	distance(p1,p2,M,p3)	Menerima parameter titik p1, p2, matriks M, dan titik p3 kemudian mengembalikan jarak dari titik p3 ke garis p1p2
5	maxDistance(p1,p2,points,M)	Menerima parameter titik p1, p2, array points yang berisi kumpulan titik, dan matriks M kemudian mengembalikan titik yang memiliki jarak terjauh dari garis p1p2 yang terdapat dalam array points
6	ConvexLeft(p1,p2,M,left)	Menerima parameter titik p1, p2, matriks M, dan boolean left kemudian apabila left bernilai True, maka fungsi mengecek titik yang berada di kiri (atau atas) garis p1p2. Jika tidak terdapat titik di kiri garis p1p2, maka p1 dan p2 membentuk pasangan titik Convex Hull ( <b>Basis dari algoritma Divide and Conquer</b> ), sebaliknya jika terdapat titik di kiri garis p1p2, maka dipilih suatu titik p3 yang memiliki jarak terjauh dari garis p1p2 kemudian dilakukan pemeriksaan lebih lanjut untuk garis p1p3 dan p3p2 ( <b>Rekursif dari algoritma Divide and Conquer</b> ). Untuk left bernilai False, proses yang terjadi sama seperti sebelumnya namun perbedaan untuk left bernilai False adalah fungsi mengecek titik yang berada di kanan (atau bawah) garis p1p2.
7	FullConvex(M)	Menerima parameter matriks M kemudian mengembalikan matriks yang berisi pasangan seluruh titik dari M yang dapat membentuk Convex Hull

No	Nama Kelas	Keterangan
----	------------	------------

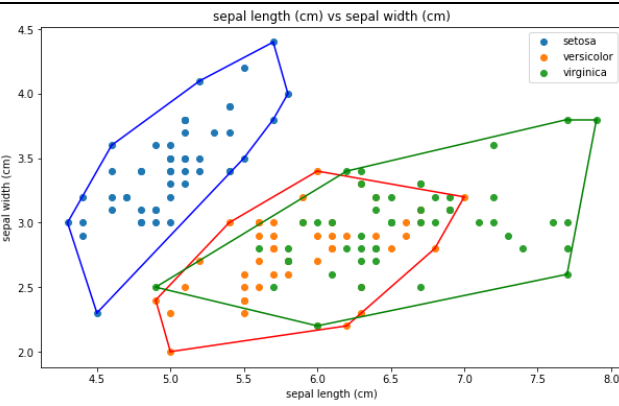
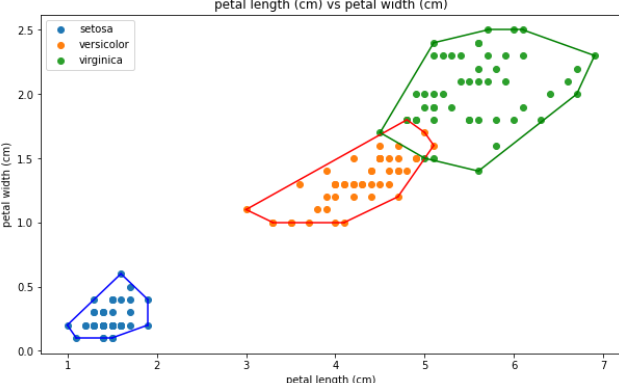
1	ConvexHull	Menerima parameter M sebagai konstruktor dari atribut simplices
---	------------	---

No	Potongan code
1	<pre>def extreme(M):     T = []     minimum = 0     maksimum = 0      for i in range(1,len(M)):         if (M[i][0] &lt; M[minimum][0] or (M[i][0] == M[minimum][0] and M[i][1] &lt; M[minimum][1])):             minimum = i         if (M[i][0] &gt; M[maksimum][0] or (M[i][0] == M[maksimum][0] and M[i][1] &lt; M[maksimum][1])):             maksimum = i     T.append(minimum)     T.append(maksimum)     return T</pre>
2	<pre>def determinan(p1,p2,array):     if ((p1[0] == p2[0] and p1[0] == array[0]) or (p1[1] == p2[1] and p1[1] == array[1])):         return 0     return (p1[0]*p2[1] + array[0]*p1[1] + p2[0]*array[1] - array[0]*p2[1] - p2[0]*p1[1] - p1[0]*array[1])</pre>
3	<pre>def leftPoints(p1,p2,M,left):     T = []      for i in range(len(M)):         if (p1 != i and p2 != i):             det = determinan(M[p1],M[p2],M[i])             if (left):                 if (det &gt; 0):                     T.append(i)             else:                 if (det &lt; 0):                     T.append(i)     return T</pre>
4	<pre>def distance(p1,p2,M,p3):     a = M[p1][1] - M[p2][1]     b = M[p2][0] - M[p1][0]     c = M[p1][0]*M[p2][1] - M[p2][0]*M[p1][1]</pre>

	<pre> jarak = abs((a*M[p3][0] + b*M[p3][1] + c)/((a**2 + b**2)**0.5)) return jarak </pre>
5	<pre> def maxDistance(p1,p2,points,M):     maksimum = points[0]      for i in range(1,len(points)):         jarak = distance(p1,p2,M,points[i])         if (jarak &gt; distance(p1,p2,M,maksimum)):             maksimum = points[i]     return maksimum </pre>
6	<pre> def ConvexLeft(p1,p2,M,Left):     T = []      if (Left):         leftpoints = leftPoints(p1,p2,M,True)         if (leftpoints == []):             T.append([p1,p2])         else:             p3 = maxDistance(p1,p2,leftpoints,M)             T1 = ConvexLeft(p1,p3,M,True)             T2 = ConvexLeft(p3,p2,M,True)             for arr in T1:                 T.append(arr)             for arr in T2:                 T.append(arr)         else:             rightpoints = leftPoints(p1,p2,M,False)             if (rightpoints == []):                 T.append([p1,p2])             else:                 p3 = maxDistance(p1,p2,rightpoints,M)                 T1 = ConvexLeft(p1,p3,M,False)                 T2 = ConvexLeft(p3,p2,M,False)                 for arr in T1:                     T.append(arr)                 for arr in T2:                     T.append(arr)      return T </pre>
7	<pre> def FullConvex(M):     Hasil = []      p1 = extreme(M)[0]     pn = extreme(M)[1] </pre>

	<pre> Hasil1 = ConvexLeft(p1,pn,M,True) Hasil2 = ConvexLeft(p1,pn,M,False)  for arr in Hasil1:     Hasil.append(arr) for arr in Hasil2:     Hasil.append(arr) Hasil = np.array(Hasil) return Hasil </pre>
8	<pre> class ConvexHull:     def __init__(this,M):         this.simplices = FullConvex(M) </pre>

### 3. Screenshoot input & output program

Input	Output																																														
<p>Pasangan atribut sepal length, sepal width (dataset iris)</p> <div><p>Dataset yang tersedia:</p><ol style="list-style-type: none"><li>iris</li><li>wine</li><li>breast cancer</li></ol><p>Silahkan memilih dataset yang diinginkan (contoh: 1): 1</p><p>(150, 5)</p><table><tr><th></th><th>sepal length (cm)</th><th>sepal width (cm)</th><th>petal length (cm)</th><th>petal width (cm)</th><th>\</th></tr><tr><td>0</td><td>5.1</td><td>3.5</td><td>1.4</td><td>0.2</td><td></td></tr><tr><td>1</td><td>4.9</td><td>3.0</td><td>1.4</td><td>0.2</td><td></td></tr><tr><td>2</td><td>4.7</td><td>3.2</td><td>1.3</td><td>0.2</td><td></td></tr><tr><td>3</td><td>4.6</td><td>3.1</td><td>1.5</td><td>0.2</td><td></td></tr><tr><td>4</td><td>5.0</td><td>3.6</td><td>1.4</td><td>0.2</td><td></td></tr></table><p>Target</p><table><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td></tr><tr><td>2</td><td>0</td></tr><tr><td>3</td><td>0</td></tr><tr><td>4</td><td>0</td></tr></table><p>Terdapat kolom-kolom sebagai berikut.</p><ol style="list-style-type: none"><li>sepal length (cm)</li><li>sepal width (cm)</li><li>petal length (cm)</li><li>petal width (cm)</li></ol><p>Silahkan memilih pasangan atribut yang diinginkan.</p><p>Kolom 1 (Contoh: 1): 1</p><p>Kolom 2 (Contoh: 2): 2</p></div>		sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	\	0	5.1	3.5	1.4	0.2		1	4.9	3.0	1.4	0.2		2	4.7	3.2	1.3	0.2		3	4.6	3.1	1.5	0.2		4	5.0	3.6	1.4	0.2		0	0	1	0	2	0	3	0	4	0	
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	\																																										
0	5.1	3.5	1.4	0.2																																											
1	4.9	3.0	1.4	0.2																																											
2	4.7	3.2	1.3	0.2																																											
3	4.6	3.1	1.5	0.2																																											
4	5.0	3.6	1.4	0.2																																											
0	0																																														
1	0																																														
2	0																																														
3	0																																														
4	0																																														
<p>Pasangan atribut petal length, petal width (dataset iris)</p>																																															



Dataset yang tersedia:

1. iris
  2. wine
  3. breast cancer
- Silahkan memilih dataset yang diinginkan (Contoh: 1): 1

```
(150, 5)
  sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm) \
0                5.1             3.5             1.4             0.2
1                4.9             3.0             1.4             0.2
2                4.7             3.2             1.3             0.2
3                4.6             3.1             1.5             0.2
4                5.0             3.6             1.4             0.2
```

```
Target
0      0
1      0
2      0
3      0
4      0
```

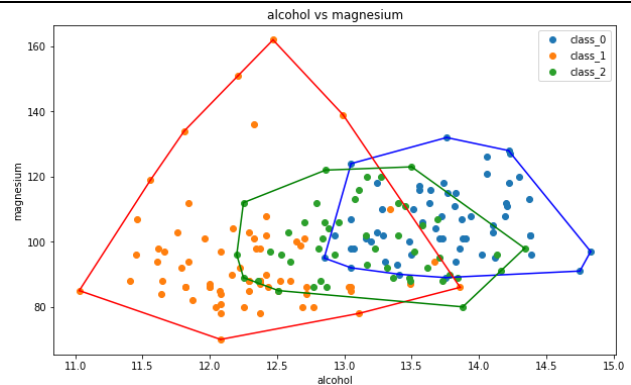
Terdapat kolom-kolom sebagai berikut.

1. sepal length (cm)
  2. sepal width (cm)
  3. petal length (cm)
  4. petal width (cm)
- Silahkan memilih pasangan atribut yang diinginkan.  
Kolom 1 (Contoh: 1): 3  
Kolom 2 (Contoh: 2): 4

## Pasangan atribut alcohol, magnesium (dataset wine)

Terdapat kolom-kolom sebagai berikut.

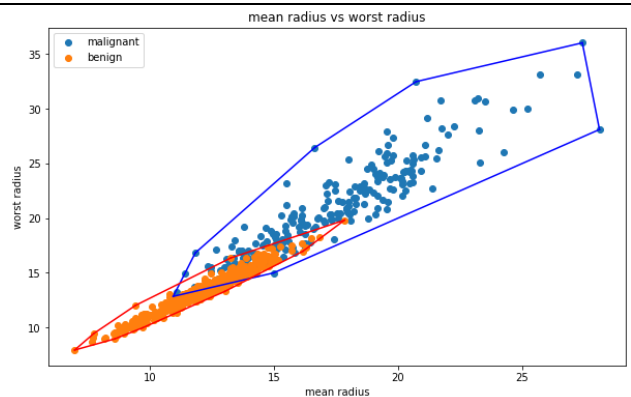
1. alcohol
  2. malic\_acid
  3. ash
  4. alcalinity\_of\_ash
  5. magnesium
  6. total\_phenols
  7. flavanoids
  8. nonflavanoid\_phenols
  9. proanthocyanins
  10. color\_intensity
  11. hue
  12. od280/od315\_of\_diluted\_wines
  13. proline
- Silahkan memilih pasangan atribut yang diinginkan.  
Kolom 1 (Contoh: 1): 1  
Kolom 2 (Contoh: 2): 5



## Pasangan atribut mean radius, worst radius (dataset breast cancer)

Terdapat kolom-kolom sebagai berikut.

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
- Silahkan memilih pasangan atribut yang diinginkan.  
Kolom 1 (Contoh: 1): 1  
Kolom 2 (Contoh: 2): 21



4. Alamat github repository kode program

<https://github.com/TubesForLyfe/Tugas-Kecil-2-Strategi-Algoritma>

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