

**Laporan Tugas Kecil 2  
IF2211 Strategi Algoritma  
Semester 2 Tahun 2021/2022**

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



**Disusun oleh**

**Gerald Abraham Sianturi**

**13520138**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2022**

Link source code: <https://github.com/geraldabrh/TucilStima02>

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. <b>Bonus:</b> program dapat menerima input dan menuliskan output untuk dataset lainnya.	✓	

## Algoritma *Divide and Conquer* yang Digunakan

Algoritma yang dipakai sebagian besar mengacu pada *slide* mata kuliah IF2211 Strategi

Algoritma *Divide and Conquer* (Bagian 4). Berikut paparannya

- Pertama, dilakukan pencarian titik terkecil dan terkanan dari sekumpulan titik-titik yang ada, misal, titik-titiknya adalah  $P_1(x_1, x_2), P_2(x_2, y_2), \dots, P_n(x_n, y_n)$ . Maka, akan dipilih titik dengan  $x_n$  terkecil (terkiri, misal  $P_a$ ) dan titik yang lain dengan  $x_n$  terbesar (terkanan, misal  $P_b$ ).
- Mengumpulkan titik-titik yang ada pada dua daerah bagi. Kumpulan titik-titik pertama merupakan titik yang berada di atas (atau kiri) garis  $P_aP_b$ , sedangkan kumpulan titik lain sebaliknya (di bawah atau kanan garis).
- Akan diproses kumpulan titik-titik pertama (daerah atas  $P_aP_b$ )
  - Jika kumpulan titik-titik pertama tidak memiliki isi, maka  $P_a$  dan  $P_b$  merupakan titik-titik yang menyusun *convex hull*. Oleh karena itu,  $P_a$  dan  $P_b$  dimasukkan ke dalam suatu NumPy Array (*arrOfConvexHull*)
  - Jika ada isi, maka akan dilakukan pencarian titik terjauh yang ada di atas garis ( $P_{max}$ ). Dan dibuat dua garis hubung, yakni  $P_aP_{max}$  dan  $P_{max}P_b$ . Jika ada titik-titik yang memiliki jarak sama ke garis acuan, akan dicari titik yang memaksimumkan sudut  $\angle P_1P_{max}P_n$
- Dicari titik-titik yang ada di atas  $P_aP_{max}$  dan  $P_{max}P_b$ . Kemudian lakukan proses poin ketiga pada kedua garis tersebut hingga kasus basis pada poin 3a terpenuhi.
- Lakukan proses yang sama pada poin ketiga dan keempat pada kumpulan titik-titik kedua (daerah bawah  $P_aP_b$ )
- Lakukan penyatuan hasil dari poin 3 dan 4 dengan poin 5
- Lakukan *ploting* terhadap setiap garis yang menghubungkan tiap ujung pada *convex hull*.

## Fungsi-fungsi dalam Modul *myConvexHull*

- findcoefconst*(point1, point2)  
Mengembalikan tiga nilai yang merupakan koefisien dan konstanta dari persamaan garis yang menghubungkan point1 dan point2. Nilai ini nanti akan digunakan pada fungsi-fungsi lainnya
- measuredistpoints*(p1, p2)  
Mengembalikan besar jarak dua titik, yakni p1 dan p2.
- measuredist*(p1, p2, pMeas)  
Mengembalikan besar jarak dari pMeas ke garis yang menghubungkan p1 dan p2
- measureAngle* (p1, p2, pRef)  
Menghitung besar sudut  $\angle P_1P_{Ref}P_2$
- lefttestRighttest*(arrOfPoint)  
Dari array dua dimensi, yang merupakan kumpulan titik berupa array 1 dimensi, akan dicari titik terkecil dan terkanan
- farthestPoint*(arrOfPoint, p1, p2)  
Dari sekumpulan array dua dimensi, akan diambil titik terjauh yang jaraknya ke garis yang menghubungkan p1 dan p2 merupakan yang terbesar
- divideArea*(arrOfPoint, p1, p2)

Mengembalikan kumpulan titik, masing-masing yang ada di atas garis yang menghubungkan p1 dan p2 dan di bawahnya

8. `convexHullParticular(arrOfPoint, lp, rp, category)`

Mengumpulkan titik-titik *convex hull* dari array dua dimensi dengan lp dan rp adalah titik-titik ujung dari garis yang akan diproses. Category berupa string yang menerima “above” untuk mengembalikan tiap titik *convex hull* di atasnya dan “below” untuk sebaliknya

9. `convexHull(arrOfPoint, lp, rp)`

Mengembalikan gabungan dari fungsi poin kesembilan dengan kategori “above” dan “below”

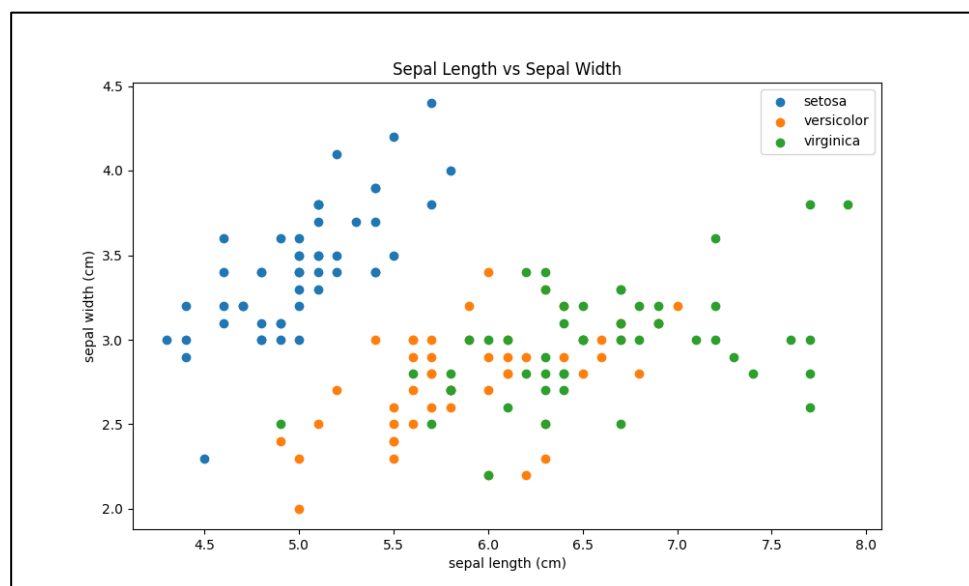
## Pengetesan

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

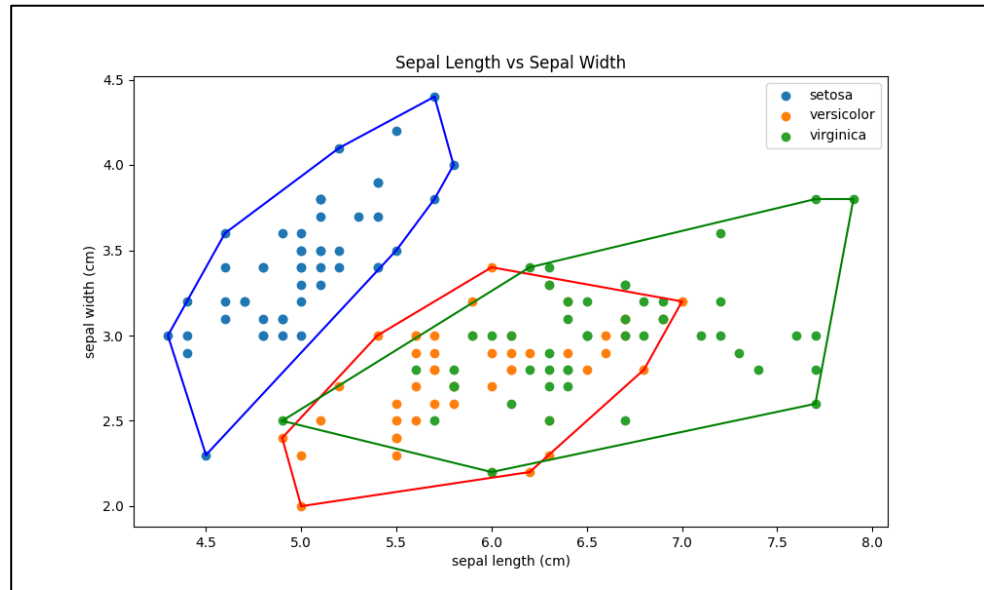
150 rows × 5 columns

Gambar 1 Data Tabular Dataset Iris

### A. Sepal Length-Sepal Width

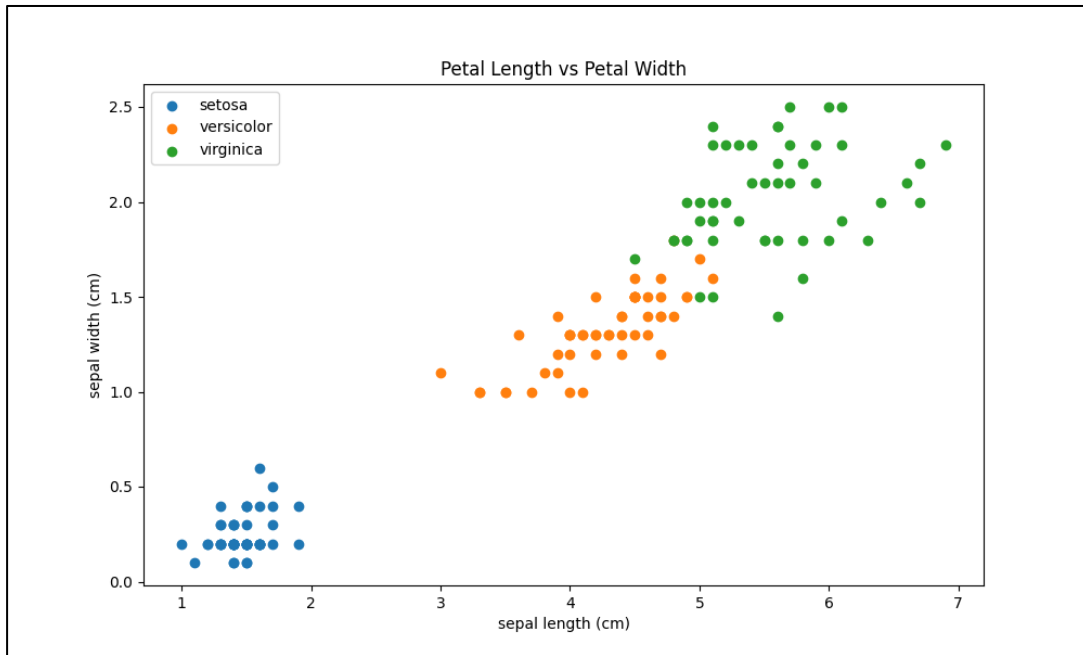


Gambar 2 Scatter Plot (Sepal Length – Sepal Width)

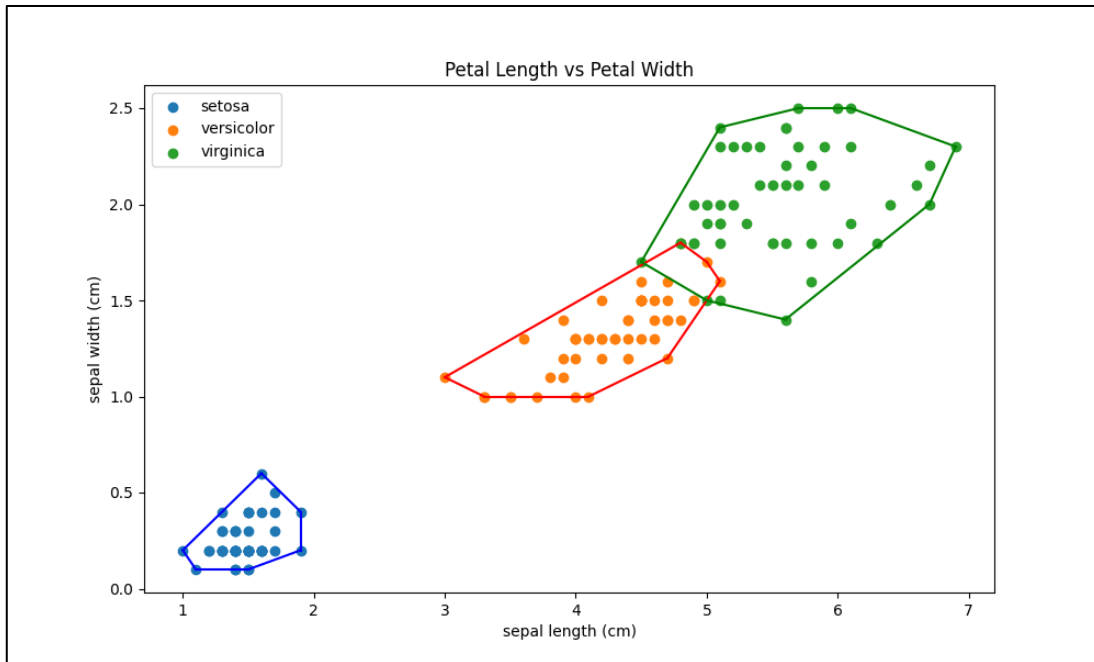


*Gambar 3 Scatter Plot (Sepal Length – Sepal Width) dengan Convex Hull*

## B. Petal Length-Petal Width



Gambar 4 Scatter Plot (Petal Length – Petal Width)



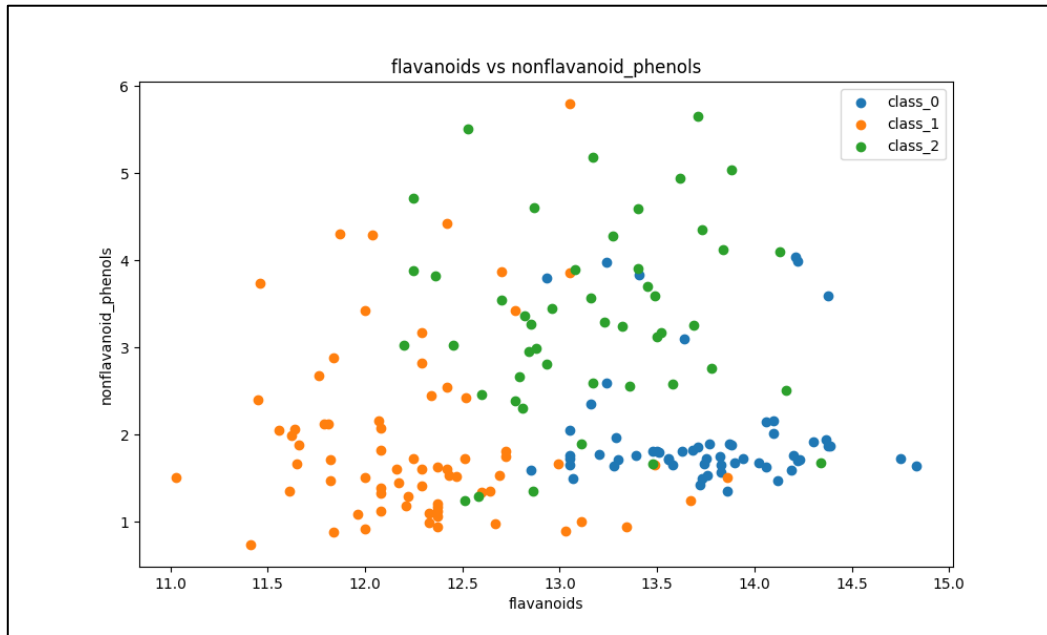
Gambar 5 Scatter Plot (Petal Length – Petal Width) dengan Convex Hull

## C. Flavonoid – nonflavanoid\_phenols pada dataset wine

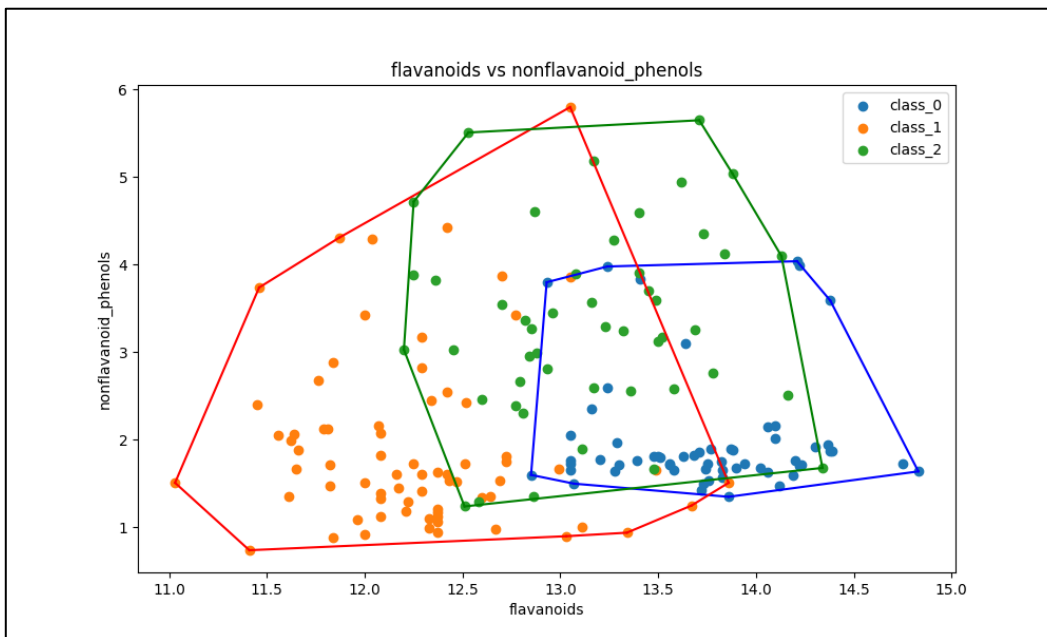
	alcohol	malic_acid	ash	alkalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	Target
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	7.70	0.64	1.74	740.0	2
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	7.30	0.70	1.56	750.0	2
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	10.20	0.59	1.56	835.0	2
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	9.30	0.60	1.62	840.0	2
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	9.20	0.61	1.60	560.0	2

178 rows × 14 columns

Gambar 6 Data Tabular Dataset Wine



*Gambar 7 Scatter Plot (flavanoids – nonflavanoid\_phenols)*



*Gambar 8 Scatter Plot (flavanoids – nonflavanoid\_phenols) dengan Convex Hull*