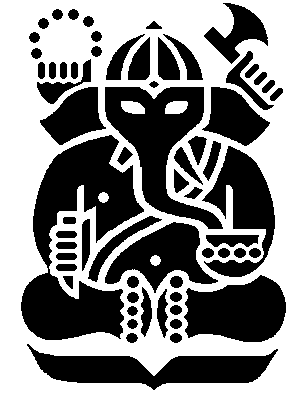
**Implementasi Convex Hull untuk Visualisasi Tes *Linear Separability Dataset* dengan Algoritma *Divide and Conquer***

LAPORAN TUGAS KECIL

Diajukan Untuk Memenuhi Tugas Kecil IF2211 Strategi Algoritma

Semester II 2021/2022



Disusun oleh:

Jason Kanggara

13520080

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

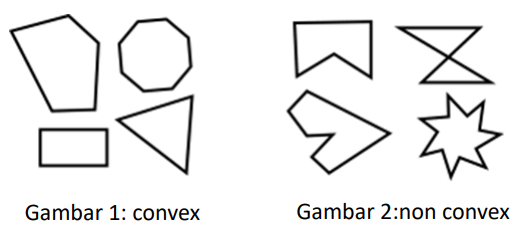
**2022**

**A. Algoritma *Divide and Conquer***

Algoritma *Divide and Conquer* adalah algoritma yang membagi suatu persoalan menjadi beberapa upa-persoalan yang memiliki kemiripan dengan persoalan semula, lalu menyelesaikan masing masing upa-persoalan secara langsung jika sudah kecil atau secara rekursif jika masih berukuran besar. Algoritma *Divide and Conquer* terbagi menjadi dua bagian, yaitu:

1. *Divide*: membagi persoalan menjadi beberapa persoalan yang lebih kecil.
2. *Conquer*: memproses upa-persoalan secara rekursif jika masih berukuran besar / langsung jika sudah berukuran kecil.

*Convex hull* merupakan salah satu hal penting dalam komputasi geometri. 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. Contoh poligon yang *convex* dan poligon yang *non-convex* dapat dilihat dari gambar berikut:



Pada tugas kali ini, mahasiswa diminta untuk membuat suatu pustaka (*library*) dalam bahasa Python yang dapat mengembalikan *convex hull* dari kumpulan data 2 dimensi (dapat dianggap kumpulan titik 2 dimensi), dengan memanfaatkan algoritma *Divide and Conquer*. Pustaka tersebut nantinya akan digunakan dalam program visualisasi data. Adapun penjelasan algoritma *Divide and Conquer* yang digunakan dalam pencarian *convex hull* pada program ini adalah sebagai berikut:

1. Dari kumpulan titik 2 dimensi yang diperoleh dari *dataset*, lakukan pengurutan untuk mencari titik ekstrim dari kumpulan titik tersebut (misal **p1**, **pn**).
2. Garis yang menghubungkan **p1** dan **pn** membagi kumpulan titik menjadi dua bagian, yaitu **s1** (kumpulan titik di sebelah kiri/atas garis **p1pn**) dan **s2** (kumpulan titik di sebelah kanan/bawah garis **p1pn**).
3. Kedua kumpulan titik tersebut (**s1** dan **s2**) akan dibagi lagi menjadi lebih kecil hingga membentuk *convex hull*.
4. Untuk sebuah bagian (misal **s1**), terdapat dua kemungkinan:
   * Jika tidak ada titik lain pada **s1**, maka titik **p1** dan **pn** menjadi pembentuk *convex hull.*
   * Jika ada titik selain **p1** dan **pn**, maka cari titik terjauh dari garis **p1pn** (misal **pmax**). Hubungkan **pmax** dengan **p1pn** maka akan terbentuk 2 garis baru.

**p1, pn,** dan **pmax** akan membentuk suatu segitiga. Titik titik yang terdapat di dalam segitiga akan diabaikan.

1. Dari 2 garis baru tersebut, kumpulkan titik yang berada di sebelah kiri garis **p1pmax** dan di sebelah kanan **pmaxpn**.
2. Ulangi langkah **4** dan **5** untuk bagian **s2** hingga bagian kiri dan kanan tidak ditemukan titik lagi.
3. Kembalikan pasangan titik yang dihasilkan.

**B. Source Code**

Bahasa pemrograman yang digunakan dalam pembuatan pustaka pencarian *convex hull* adalah Python

**C. Hasil Program**

|  |
| --- |
|  |
|  |
|  |

**D. Link Source Code**

[*https://github.com/jasonk19/Tucil-Stima-2.git*](https://github.com/jasonk19/Tucil-Stima-2.git)

**E. Tabel Cek List**

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Pustaka *myConvexHull* berhasil dibuat dan tidak ada kesalahan | √ |  |
| 1. *Convex hull* yang dihasilkan sudah benar | √ |  |
| 1. Pustaka *myConexHull* dapat digunakan untuk menampilkan *convex hull* setiap label dengan warna yang berbeda | √ |  |
| 1. **Bonus:** program dapat menerima input dan menuliskan output untuk dataset lainnya | √ |  |