

## 1. K-means Clustering

- Kelebihan :
  1. Mudah diimplementasikan
  2. Waktu yang dibutuhkan untuk pembelajaran cepat
  3. Sistem yang fleksibel dan mudah beradaptasi
  4. Dapat dijabarkan tanpa statistic karena prinsipnya sederhana
- Kekurangan :
  1. Perlu menginisialisasikan nilai K secara random terlebih dahulu
  2. Tidak dapat meng-handle data dengan density dimensi yang tinggi dengan baik
  3. Perhitungan jika data banyak akan memakan waktu
  4. Penentuan nilai optimal susah didapatkan dengan K random
- Cara Kerja :
  1. Tentukan K yang akan digunakan
  2. Inisialisasi posisi K tersebut secara random (kedepannya disebut centroid)
  3. Untuk setiap dataset yang ada, cari centroid mana yang paling dekat
  4. Hitung rerata dari masing-masing data ke centroid
  5. Update posisi centroid
  6. Lakukan step 3 s/d 5 sampai posisi centroid tidak berubah, lalu hentikan iterasi

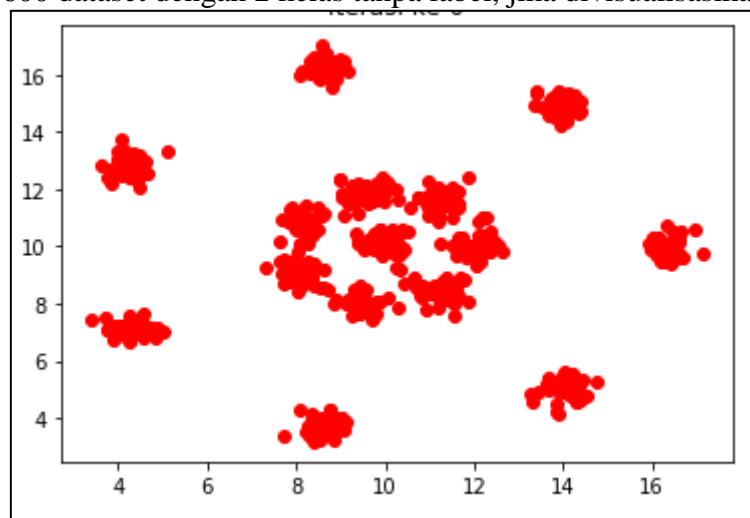
## 2. Agglomerative Clustering :

1. Menghitung Matriks jarak antar data
2. Menggabungkan dua kelompok terdekat berdasarkan parameter yang ditentukan
3. Perbarui matrik jarak antar data untuk merepresentasikan kedekatan antara kelompok baru dan yang masih tersisa
4. Ulang step 2 dan 3 hingga hanya tersisa 1 kelompok

## 3. Self-Organizing Maps :

Analisa :

- Terdapat 600 dataset dengan 2 kelas tanpa label, jika divisualisasikan :

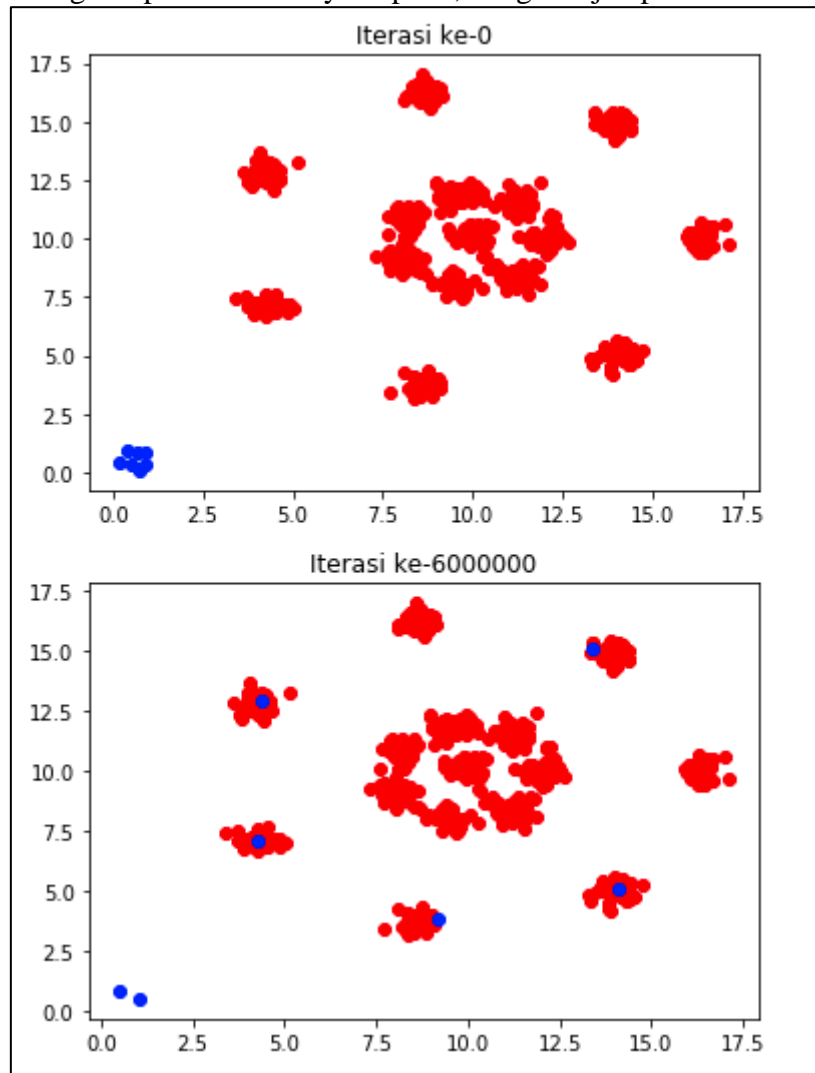


- Nilai maksimal kelas pertama adalah 17.124, dan nilai minimumnya adalah 3.402
- Nilai maksimal kelas kedua adalah 17.012, dan nilai minimumnya adalah 3.178
- Inisialisasi neuron secara random berisikan nilai float antara 0 hingga 1

Penyelesaian Masalah:

1. Membangkitkan X neuron dan inisialisasi Weightnya, dalam kasus ini  $X = 7$
2. Tentukan  $\text{learningRate} = 0.1$ ,  $\text{learningStep} = 2$ ,  $\text{Sigma} = 2$ ,  $\text{sigmaStep} = 2$
3. Kemudian tentukan epoch sebagai penentu berapa banyak iterasi yang akan dilakukan

4. Lalu lakukan perhitungan untuk setiap neuron dengan semua dataset yang ada
  - a. Tahap yang dilakukan adalah, menghitung BMU dari semua neuron terhadap suatu data didalam dataset
  - b. Ambil nilai BMU (nilai minimum dari hasil perhitungan menggunakan Euclidean)
  - c. Ambil index dari neuron yang merupakan nilai BMU, dan lakukan perhitungan neuron tetangganya (menghitung  $DW_i$ , dan  $XW_i$ )
  - d. Lalu perbarui weight neuron dengan perhitungan : Weight Neuron saat ini –  $Dw_i$
  - e. Ulangi Step diatas sebanyak epoch, dengan laju epoch = +1



Setelah menjalankan program sebanyak epoch maka beberapa titik neuron sudah bergerak menuju pengelompokannya masing masing pada iterasi ke 600.000  
Penambahan jumlah epoch dapat mengoptimalkan hasil dari SOM