

Nama : Irvan Naufali Rahmanto

NIM : 1301164062

Kelas : IF 40 08

Matakuliah : Machine Learning

## 1. *K-Means Clustering*

Definisi dari *k-Means Clustering* merupakan salah satu metode algoritma clustering, dimana bertujuan untuk membagi data menjadi beberapa kelompok dengan meminimasi *Sum of Squared Error* (SSE) antara objek objek data dengan sejumlah *k centroid*, algoritma ini menerima masukan berupa data tanpa label kelas, atau biasa disebut dengan *unsupervised learning*.

- **Kelebihan :**

- Mudah diimplementasikan dan dijalankan
- Waktu yang dibutuhkan untuk menjalankan *relative* lebih mudah dan cepat
- Mudah untuk diadaptasi
- Umum dan familiar saat digunakan
- Sesuai untuk himpunan data teratur atau bulat

- **Kekurangan :**

- Sebelum algoritma dijalankan, *k* buah titik akan diinisialisasi secara random hingga pengelompokkan berbeda-beda, jika nilai random inisialisasi tidak sesuai maka akan berpengaruh pada pengelompokkan yang dihasilkan tidak akurat.
- Ukuran cluster tidak seimbang (*unbalanced*), artinya jumlah data di tiap cluster sangat jauh berbeda
- Kurang sesuai untuk himpunan data tidak teratur

- **Contoh :**

Pada contoh representative *k-means clustering* pengelompokan untuk pemahaman bertujuan menghasilkan kelompok kelompok yang terdiri dari objek objek dengan karakteristik yang serupa, seperti hal nya manusia mengelompokkan objek objek. Contoh nya digunakan untuk melakukan segmentasi pasar, adalah pengelompokkan pelanggan sesuai karakteristik mereka (misal : gaya hidup, kebutuhan), dapat digunakan dalam pemberi rekomendasi untuk pengelompokkan objek yang saling terkait juga.

## 2. Agglomerative Hierarcal Clustering

Pada dasarnya konsep dasar dari *Agglomerative Hierarcal Clustering* adalah sebuah klastering hirarki untuk mengelompokkan objek objek data k dalam hirarki klaster dengan perhitungan (*Bottom-up*), maksud nya ialah bekerja dengan menganggap tiap obkel tunggal sebagai sebuah klaster kemudian secara iterative menggabungkan sepasang klaster yang paling dekat/mirip menjadi klaster besar, sampai dihasilkan sebuah klaster besar yang berisi semua objek data, strategi ini lebih mudah diimplementasikan sehingga banyak digunakan dalam aplikasi praktis.

- **Contoh :**

Sebagai contoh nya, pada himpunan data *handphone* , kita bisa menemukan sebuah struktur bertingkat terkait dengan kekuatan baterai, ada kelompok *handphone* dengan kekuatan baterai 12 jam hingga 20 jam missal nya, sedangkan kelompok lain nya memiliki kekuatan baterai 21 jam hingga 28 jam.

## 3. Laporan Self Organizing Map

### A. Analisis masalah

Analisa yang terdapat pada masalah ini ialah disana terdapat sebuah dataset berisi 600 objek data yang memiliki dua atribut tanpa label kelas. Dari permasalahan yang didapatkan kita dimaksimalkan untuk membangun sebuah metode model klasterisasi atau (*clustering*) dengan metode cara *Self Organizing Map* yang bertujuan menghasilkan sejumlah klaster yang paling optimum nanti nya.

### B. Strategi penyelesaian

#### a) Melakukan penelusuran menghitung jarak

Penelusuran pada jarak yang dilakukan dimetode SOM ini menggunakan rumus *Euclidian* dimana perhitungan jarak nya berdasarkan objek data dengan weight awal atau neuron yang berada pada dataset. Pada permasalahan ini rumus *Euclidian* nya adalah  $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$  dengan O pada program berarti objek.

#### b) Perhitungan jarak terdekat neuron dengan tetangga

Perhitungan jarak di step kali ini dengan menggunakan rumus *Euclidian* sama seperti sebelum nya, dimana perhitungan ini bertujuan agar mengetahui  $S_n$  neuron dengan tetangga nya.

**c) Pencarian  $T_n$  dari jarak terdekat neuron tetangga**

$T_n$  pada jarak yang akan dicari ini meliputi hasil pencarian pada step sebelum nya, dimana pencarian didasarkan dengan rumus  $T_n = \exp(-S_n \ln 2) / 2 * \tau \sigma^2$  dari rumus pencarian  $T_n$  dengan eksponen tersebut, dapat diinformasikan bahwa  $\tau \sigma$  disana bernilai 2, sesuai dengan perintah nya.

**d) Perhitungan  $\Delta W_n$  sesuai objek dan neuron nya**

Perhitungan nilai  $W$  sesuai objek dan neuron nya meliputi hasil dari pencarian  $T_n$  di step sebelum nya, dimana nanti akan menggunakan rumus perhitungan  $Learning\ rate (\tau \eta) * T_n * (objek\ n - Neuron\ n)$ . *learning rate* disana bernilai 0.1.

**e) Melakukan pembaruan pada unit pemenang**

Pembaruan dimaksudkan dengan melakukan perulangan dengan iterasi yang terbaru dari dataset awal pada objek hingga dataset yang paling akhir, perhitungan juga didasarkan pada pengerjaan step sebelum nya, yaitu dengan rumus  $N^{\wedge} = N + \Delta W_{On}$ , disini akan menghasilkan neuron atau weight yang baru dari perhitungan iterasi tersebut.

**C. Hasil Output**

**a) Hasil clustering**

```
[[2.175185851361436, 9.587044394352406], [14.140819685029662, 6.780693330073094], [0.04794366090733693, 11.110973811563396]]
n terakhir
[[14.97518585136141, 8.287044394352407], [15.240819685029658, 0.9806933300730769], [14.947943660907315, 0.21097381156337747]]
```