1. Kelebihan :
   - pada saat pengimplementasian dan di jalankan k-means lebih mudah
   - waktu pembelajaran lebih cepat
   - k-means sangat umum penggunaannya
   - menggunakan prinsip yang sederhana

   Kekurangan:
   - K diinisialisasikan secara random sehingga pengelompokan data dapat berbeda-beda
   - Pencarian titik terdekat dengan K memakan waktu yang lama jika menggunakan banyak data
   - Penggunaan k buah random tidak ada jaminan untuk menemukan kumpulan cluster yang optimum

   Contoh kasus:
   - Menentukan banyaknya cluster misalnya k = 2 dan banyaknya cluster harus lebih kecil dari banyaknya data

n	a	b
1	1	1
2	2	1
3	4	3
4	5	4

Contoh Dataset K-means

inisialisai centorid dataset pada tabel dataset adalah C1 = {1,1} dan C2 = {2,1}. centroid awal dapat ditentukan manual ataupun random.

Perulangan berikutnya (perulangan ke-1 sampai selesai) maka lakukan perhitunagn nilai rata-rata data pada setiap cluster untuk mendapatkan centroid baru. Proses clustering akan selesai jika centroid yang baru dihitung sama dengan sebelumnya jika berbeda maka proses akan tetap lanjut ke langkah berikutnya

untuk menghitung jarak menggunakan euclidiean distance.

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

Perulangan ke-1 :

	a	b
c1	1	1
c2	3.666667	2.666667

Nilai Rata-Rata Centroid pada Pengulangan ke-1

Perulangan ke-3 :

	a	b
c1	1.5	1
c2	4.5	3.5

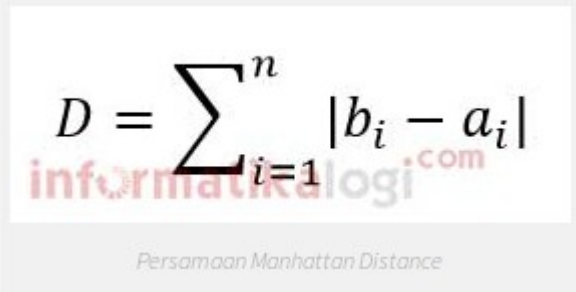
Nilai Rata-Rata Centroid pada Pengulangan ke-3

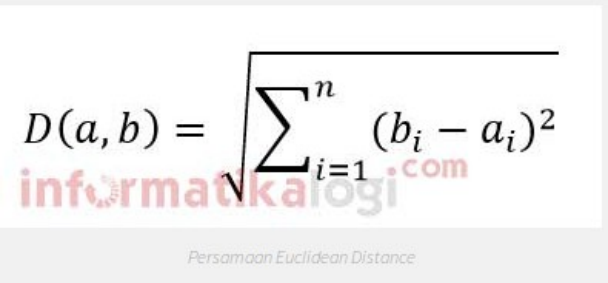
Perulangan ke-2 :

	a	b
c1	1.5	1
c2	4.5	3.5

Nilai Rata-Rata Centroid pada Pengulangan ke-2

2. Konsep dasar Agglomerative Hierarchical Clustering (bottom-ip) yaitu dengan menggabungkan beberapa buah klaster menjadi satu klaster tunggal, dimulai dengan meletakkan setiap objek data sebagai sebuah klaster tersendiri, selanjutnya klaster-klaster tersebut di gabungkan menjadi klaster yang lebih besar sampai akhirnya semua objek data menjadi satu dalam sebuah klaster tunggal. Menghitung matrik jarak menggunakan **Manhattan Distance** atau pun dengan **Euclidian Distance**





Berikut metode pengelompokan **Agglomerative Hierarchical :**

a) **Single Linkage (Jarak Terdekat)**

b) **Complete Linkage (Jarak Terjauh)**

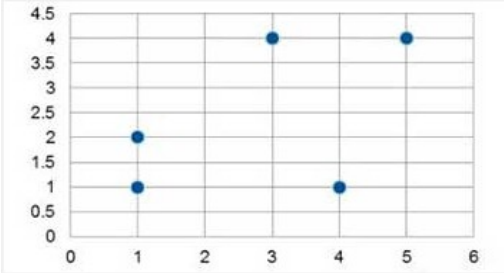
c) **Average Linkage (Jarak Rata-Rata)**

contoh kasus dan ilustrasi yang representatif :

Data	Fitur x	Fitur y
1	1	1
2	4	1
3	1	2
4	3	4
5	5	4

dataset-hierarchical-clustering

Kelompokkan dataset tersebut dengan menggunakan metode AHC (Single Linkage, Complete Linkage dan Average Linkage) menggunakan jarak Manhattan



dataset-visual-hierarchical-clustering

Menghitung jarak pada semua pasangan dua data:

$$D = \sum_{i=1}^{n} |b_i - a_i|$$

Persamaan Manhattan Distance

D_{man} (*Data1*, *Data1*) = |1-1| + |1-1| = 0
D_{man} (*Data1*, *Data2*) = |1-4| + |1-1| = 3
D_{man} (*Data1*, *Data3*) = |1-1| + |1-2| = 1
D_{man} (*Data1*, *Data4*) = |1-3| + |1-4| = 5
D_{man} (*Data1*, *Data5*) = |1-5| + |1-4| = 7
D_{man} (*Data2*, *Data3*) = |4-1| + |1-2| = 4
D_{man} (*Data2*, *Data4*) = |4-3| + |1-4| = 4
D_{man} (*Data2*, *Data5*) = |4-5| + |1-4| = 4
D_{man} (*Data3*, *Data4*) = |1-3| + |2-4| = 4
D_{man} (*Data3*, *Data5*) = |1-5| + |2-4| = 6
D_{man} (*Data4*, *Data5*) = |3-5| + |4-4| = 2

Dman	1	2	3	4	5
1	0	3	1	5	7
2	3	0	4	4	4
3	1	4	0	4	6
4	5	4	4	0	2
5	7	4	6	2	0

Dman-hierarchical-clustering

- **Salah satu contoh menggunakan metode single link :**

  Dengan memperlakukan data sebagai kelompok, selanjutnya kita pilih jarak dua kelompok yang terkecil. $\min(D_{man}) = \min(d_{13}) = 1$
  Terpilih kelompok 1 dan 3, sehingga kedua kelompok ini digabungkan. Menghitung jarak antar kelompok (1 dan 3) dengan kelompok lain yang tersisa, yaitu 2, 4 dan 5.

  $d_{(13)2} = \min \{d_{12}, d_{32}\} = \min \{3,4\} = 3$
  $d_{(13)4} = \min \{d_{14}, d_{34}\} = \min \{5,4\} = 4$
  $d_{(13)5} = \min \{d_{15}, d_{35}\} = \min \{7,6\} = 6$

  Dengan menghapus baris-baris dan kolom-kolom matrik jarak yang bersesuaian dengan kelompok 1 dan 3, serta menambahkan baris dan kolom untuk kelompok (13).Selanjutnya dipilih jarak dua kelompok yang terkecil. $\min(D_{man}) = \min(d_{45}) = 2$ Menghitung jarak antar kelompok (4 dan 5) dengan kelompok lain yang tersisa, yaitu (13) dan 2. $d_{(45)(13)} = \min \{d_{41}, d_{43}, d_{51}, d_{53}\} = \min \{5,4,7,6\} = 4$, $d_{(45)2} = \min \{d_{42}, d_{52}\} = \min \{4,4\} = 4$

  Menghapus baris dan kolom matrik yang bersesuaian dengan kelompok 4 dan 5, serta menambahkan baris dan kolom untuk kelompok (45). Selanjutnya dipilih jarak dua kelompok yang terkecil. $\min(D_{man}) = \min(d_{(13)2}) = 3$ Terpilih kelompok (13) dan 2, sehingga kedua kelompok ini digabungkan. (Melanjutkan pengelompokan). Menghitung jarak antar kelompok ((13) dan 2) dengan kelompok lain yang tersisa, yaitu (45). $d_{(132)(45)} = \min \{d_{14}, d_{15}, d_{34}, d_{35}, d_{24}, d_{25}\} = \min \{5,7,4,6,4,4\} = 4$.

  Menghapus baris dan kolom matrik yang bersesuaian dengan kelompok (13) dan 2, serta menambahkan baris dan kolom untuk kelompok (123). Jadi kelompok (132) dan (45) digabung untuk menjadi kelompok tunggal dari lima data, yaitu kelompok (13245) dengan jarak terdekat 4.

3. **Problem :**

Diberikan sebuah dataset berisi 600 objek data yang memiliki dua atribut tanpa label kelas. Bangunlah sebuah model klasterisasi (*clustering*) menggunakan metode *Self Organizing Map* (SOM) untuk menghasilkan sejumlah klaster yang paling optimum.

Strategi Penyelesaian Masalah :

Strategi yang digunakan dengan mengimplementasikan metode *Self Organizing Map* kedalam program sehingga program tersebut dapat menentukan label dari 600 data yang diberikan. Dalam program yang saya buat, saya membangkitkan neuron sebanyak 10 neuron dengan weight dari masing-masing neuron di random dari -15 sampai 15 di awal penentuan neuronnya, setelah itu setiap data akan di hitung jaraknya ke masing-masing neuron dan neuron yang terdekat dari data tersebut akan di pilih menjadi neuron pemenang, Stelah itu akan di hitung jarak antar neuronnya ke neuron pemenang jika jaraknya lebih kecil dari 10 maka neuron tersebut akan di pilih menjadi neuron tetangganya, untuk menghuitng jarak menggunakan ***Euclidian Distance*** , Setelah mendapatkan jarak dari antar neuron maka cari Tnnya dengan rumus EXP(-(Sn^2)/2*sigma), setelah itu akan dicari wnnya dengan rumus Lr*Tn*(x-n), dimana Lr dan sigma adan selalu berubah di setiap iterasinya dan terakhir maka mengubah nilai weight dari neuronnya, setelah itu mengkelompokan data ke neuron terdekatnya.

Analisis :

Berdasarkan penjelasan yang saya jabarkan maka dapat diketahui bahwa pembangkitkan berapa banyak neuron sangat penting untuk mendapatkan klaster yang optimum, dan pengambilan tetangga dari neuron pemenangnya juga sangat berperan penting.

Hasil Percobaan :