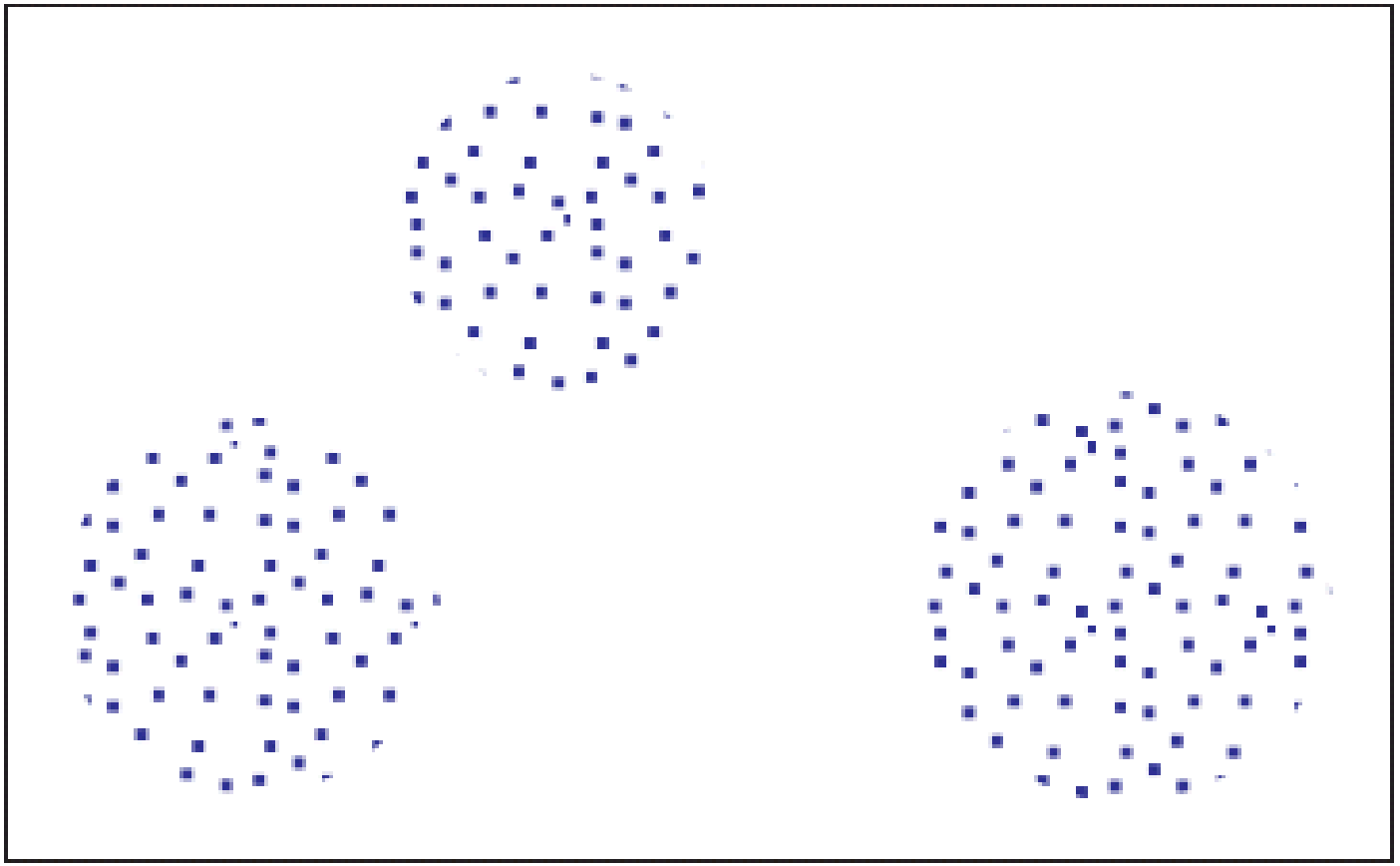
DATA MINING DAN WAREHOUSE

ANDRI

CLUSTERING

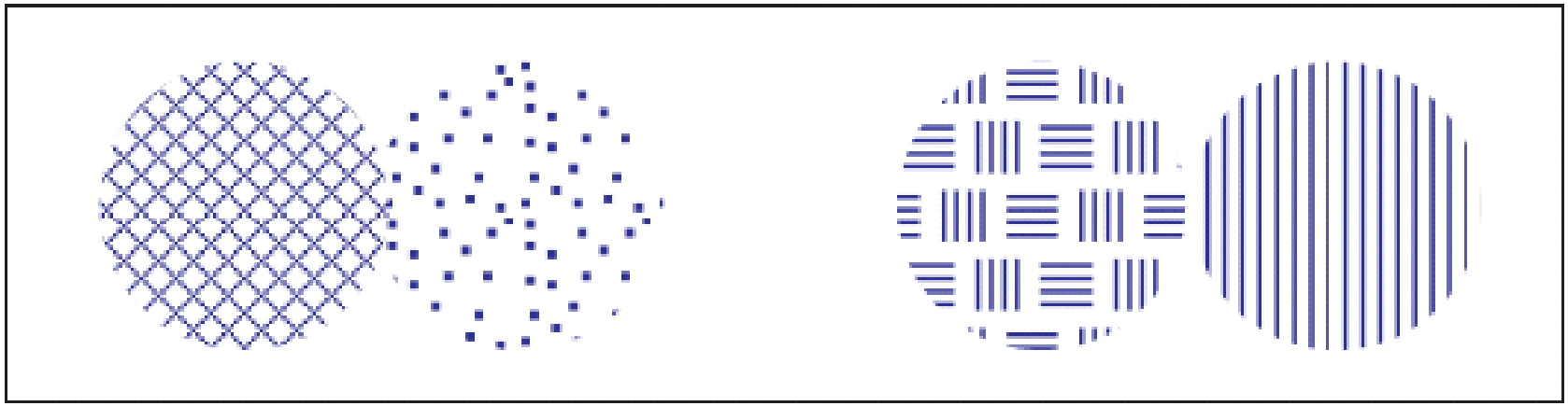
- Secara umum *cluster* didefinisikan sebagai "*sejumlah objek yang mirip yang dikelompokan secara bersama*",
- Namun definisi dari *cluster* bisa beragam tergantung dari sudut pandang yang digunakan,
- beberapa definisi *cluster* berdasarkan sudut pandang adalah sebagai berikut :

- Definisi *Well-Separated Cluster*
- Berdasarkan definisi ini *cluster* adalah sekelompok titik(objek) dimana sebuah titik pada kelompok itu lebih dekat atau mirip dengan semua titik(objek) yang ada pada kelompok tersebut dari pada titik-titik (objek-objek) lain yang tidak terdapat pada kelompok itu. Biasanya digunakan sebuah nilai batas (*threshold*) untuk menentukan titik-titik (objek-objek) yang dianggap cukup dekat satu sama lainnya. Namun terdapat kelemahan pada definisi ini yaitu titik-titik yang terdapat pada "pojok" sebuah *cluster* pada kenyataannya mungkin saja lebih dekat dengan titik-titik pada *cluster* yang lain.



Cluster* berdasarkan definisi *Well-Separated-Cluster

- Definisi *Center-Based Cluster*
- Berdasarkan definisi ini sebuah *cluster* didefinisikan sebagai sekelompok titik (objek) dimana semua titik pada kelompok itu lebih dekat dengan pusat atau "*center*" dari kelompok tersebut dari pada pusat pada kelompok lainnya.
- Umumnya pusat *cluster* adalah *centroid*, yaitu rata-rata dari semua titik pada *cluster* tersebut, namun dapat juga digunakan *medoid*, yaitu titik yang paling mewakili pada sebuah *cluster*.



Cluster* berdasarkan definisi *Center-Based Cluster

Cluster Analysis

- *Cluster analysis* merupakan salah satu metode *Data mining* yang bersifat tanpa latihan (*unsupervised analisys*) yang mempunyai tujuan untuk mengelompokan data kedalam kelompok-kelompok dimana data-data yang berada dalam kelompok yang sama akan mempunyai sifat yang relatif *homogen*.

- Jika ada *n* objek pengamatan dengan *p* variable maka terlebih dulu ditentukan ukuran kedekatan sifat antar data, ukuran kedekatan sifat data yang bisa digunakan adalah jarak euclidius (*Euclidean distance*) antara dua objek dari *p* dimensi pengamatan, jika objek pertama yang akan diamati adalah X = $[x_1,x_2,x_3,....x_p]$ dan Y=$[y_1,y_2,y_3,....y_p]$ maka *euclidean distance* dirumuskan sebagai berikut :

$$d_{(X,Y)} = \sqrt{\sum_{j=1}^{p}(x_j - y_j)^2}$$

- Secara formal definisi dari *cluster analysis* adalah sebagai berikut:
- Misalkan *S* adalah himpunan objek yang mempunyai *n* buah elemen,

$$S = \{o_1, o_2, o_3 \ldots o_n\} \qquad \text{(II.1)}$$

- *Cluster analysis* membagi *S (didefinisikan pada persamaan II.1)* menjadi *k* himpunan $C_1, C_2, C_3 \ldots C_k$, himpunan-himpunan tersebut disebut dengan *cluster*. Sebuah *cluster* C_i adalah subset atau himpunan bagian dari *S*, . Solusi atau keluaran dari sebuah *cluster Analysis* dinyatakan sebagai himpunan dari semua *cluster*,

- Jika S adalah himpunan objek yang mempunyai *n* buah elemen dan terdiri dari *r* variable maka ketika S dibagi menjadi *k cluster*, maka model dari *cluster* dapat didefinisikan dengan dua buah matrik yaitu matrik data $D_{nxk} = (d_{ik})$ dan matrik variable $F_{rxk} = (f_{jk})$,

$$d_{ik} = \begin{cases} 1, \text{ data ke i anggota kluster ke k} \\ 0, \text{data ke i bukan anggota kluster ke k} \end{cases}$$

- Proses *clustering* mengasumsikan bahwa data akan menjadi anggota dari satu dan hanya satu *cluster*.

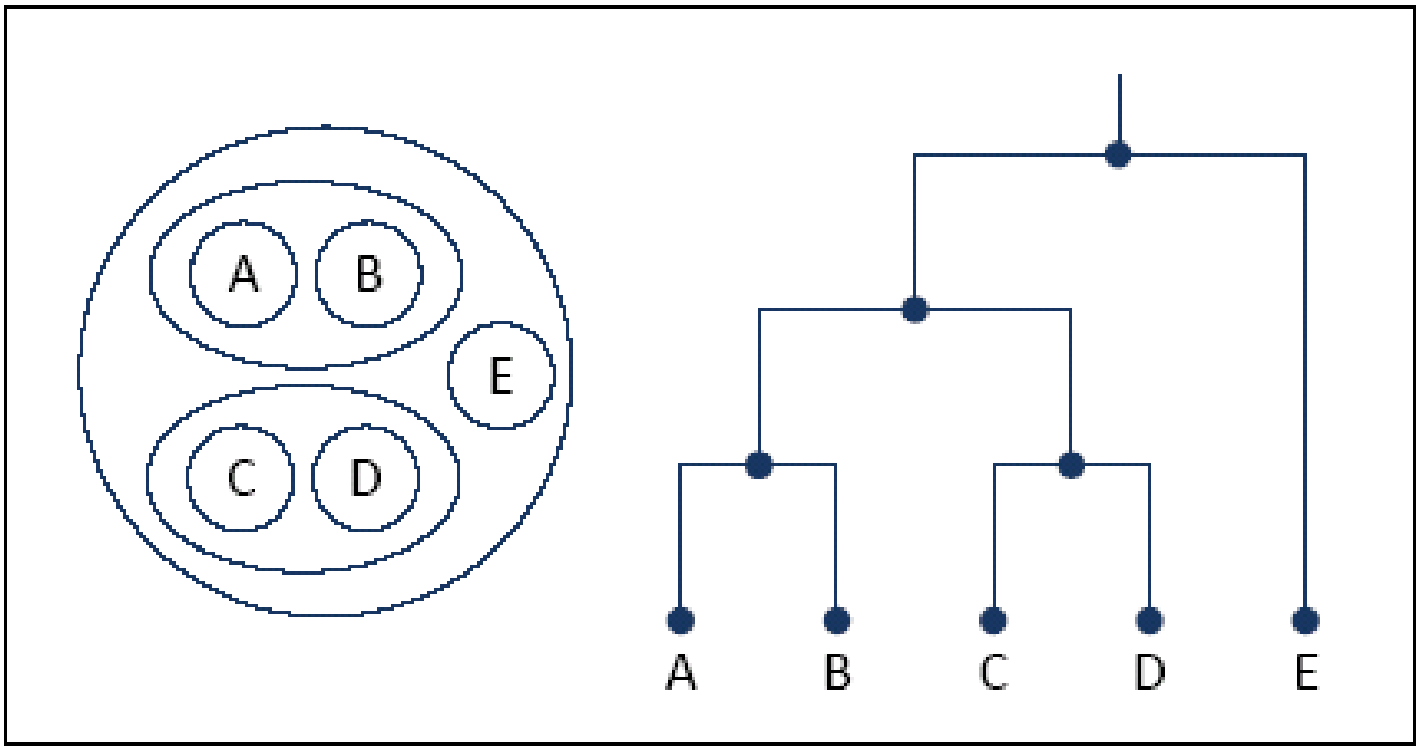
$$f_{jk} = \begin{cases} 1, \text{ Variable ke j anggota kluster ke k} \\ 0, \text{Variable ke j bukan anggota kluster ke k} \end{cases}$$

Klasifikasi Metode *Cluster Analysis*

- Metode *cluster analysis* pada dasarnya ada dua jenis, yaitu metode *cluster analysis* hirarki (*hierarchical clustering method*) dan Metode *cluster analysis* non hirarki (*non hierarchical clustering method*).

Metode *clustering* hirarki

- Metode *clustering* hirarki digunakan apabila *belum ada informasi jumlah cluster* yang akan dipilih, metode hirarki akan menghasilkan *cluster-cluster* yang bersarang (*nested*) sehingga masing-masing *cluster* dapat memiliki sub-*cluster*.
- Prinsip utama metode *cluster analysis* hirarki adalah mengatur semua objek dalam sebuah pohon keputusan (umumnya berupa pohon biner) berdasarkan suatu fungsi kriteria tertentu. Pohon tersebut disebut dendogram.



Contoh Dendogram

- Semakin tinggi level simpul pohon maka semakin rendah tingkat similaritas antar objeknya, metode *cluster analysis* hirarki dapat dilakukan dengan dua pendekatan yaitu *bottom-up* (*agglomerative*) dan *top-down* (*divisive*).
- Pada pendekatan *aggromerative* setiap objek pada awalnya berada pada *cluster* masing-masing, kemudian setiap *cluster* yang paling mirip akan dikelompokan dalam satu *cluster*, hingga membentuk suatu hirarki *cluster*.

- Pada pendekatan *divisive*, pada awalnya hanya terdapat satu buah *cluster* tunggal yang beranggotakan seluruh objek, kemudian dilakukan pemecahan atas *cluster* tersebut menjadi beberapa sub-*cluster*, contoh algoritma metode *cluster* hirarki adalah HAC (*Hiearchical Aggromerative Clustering*) dengan beberapa variasi perhitungan similaritas antar *cluster* seperti *single-link*, *complete-link* dan *group average*.

Metode *Cluster Analysis* Non Hirarki

- Metode *cluster analysis* non hirarki biasa juga disebut dengan *partitional clustering* bertujuan mengelompokan *n* objek kedalam *k cluster* ($k < n$) dimana *nilai k sudah ditentukan sebelumnya*.
- Salah satu prosedur *clustering* non hirarki adalah menggunakan metode *K-Means clustering analisis*, yaitu metode yang bertujuan untuk mengelompokan objek atau data sedemikian rupa sehingga jarak tiap objek ke pusat *cluster* (*centroid*) adalah minimum, titik pusat *cluster* terbentuk dari rata-rata nilai dari setiap variable.

- Secara umum proses *cluster analysis* dimulai dengan perumusan masalah *clustering* dengan mendefinisikan variable-variable yang akan digunakan sebagai dasar proses *cluster*.
- Konsep dasar dari *cluster analysis* adalah konsep pengukuran jarak (*distance*) atau kesamaaan (*similarity*),
- distance adalah ukuran tentang jarak pisah antar objek sedangkan similaritas adalah ukuran kedekatan.
- Pengukuran jarak (*distance type measure*) digunakan untuk data-data yang bersifat metrik, sedangkan pengukuran kesesuaian (*matching type measure*) digunakan untuk data-data yang bersifat kualitatif atau non metrik.

- Proses *clustering* yang baik seharusnya menghasilkan *cluster-cluster* yang berkualitas tinggi dengan sifat-sifat sebagai berikut :

1. Setiap objek pada *cluster* memiliki kemiripan (*intra cluster similarity*) yang tinggi satu sama lainnya.
2. Kemiripan objek pada *cluster* yang berbeda(*inter cluster similarity*) rendah.

K-Means Cluster Analysis

- *K-means cluster analysis* merupakan salah satu metode *cluster analysis* non hirarki yang berusaha untuk mempartisi data yang ada kedalam satu atau lebih *cluster* atau kelompok data berdasarkan karakteristiknya,
- sehingga data yang mempunyai karakteristik yang sama dikelompokan dalam satu *cluster* yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokan ke dalam *cluster* yang lain.
- Tujuannya adalah untuk meminimalkan *objective function* yang di set dalam proses *clustering*, yang pada dasarnya berusaha untuk meminimalkan variasi dalam satu *cluster* dan memaksimalkan variasi antar *cluster*.

- K-Means meliputi *sequential threshold, pararel threshold dan optimizing threshold*
- *Sequential threshold* melakukan pengelompokan dengan terlebih dahulu memilih satu objek dasar yang akan dijadikan nilai awal *cluster*, kemudian semua *cluster* yang ada dalam jarak terdekat dengan *cluster* ini akan bergabung, lalu dipilih *cluster* kedua dan semua objek yang mempunyai kemiripan dengan *cluster* ini akan digabungkan, demikian seterusnya sehingga terbentuk beberapa *cluster* dengan keseluruhan objek terdapat didalamnya.

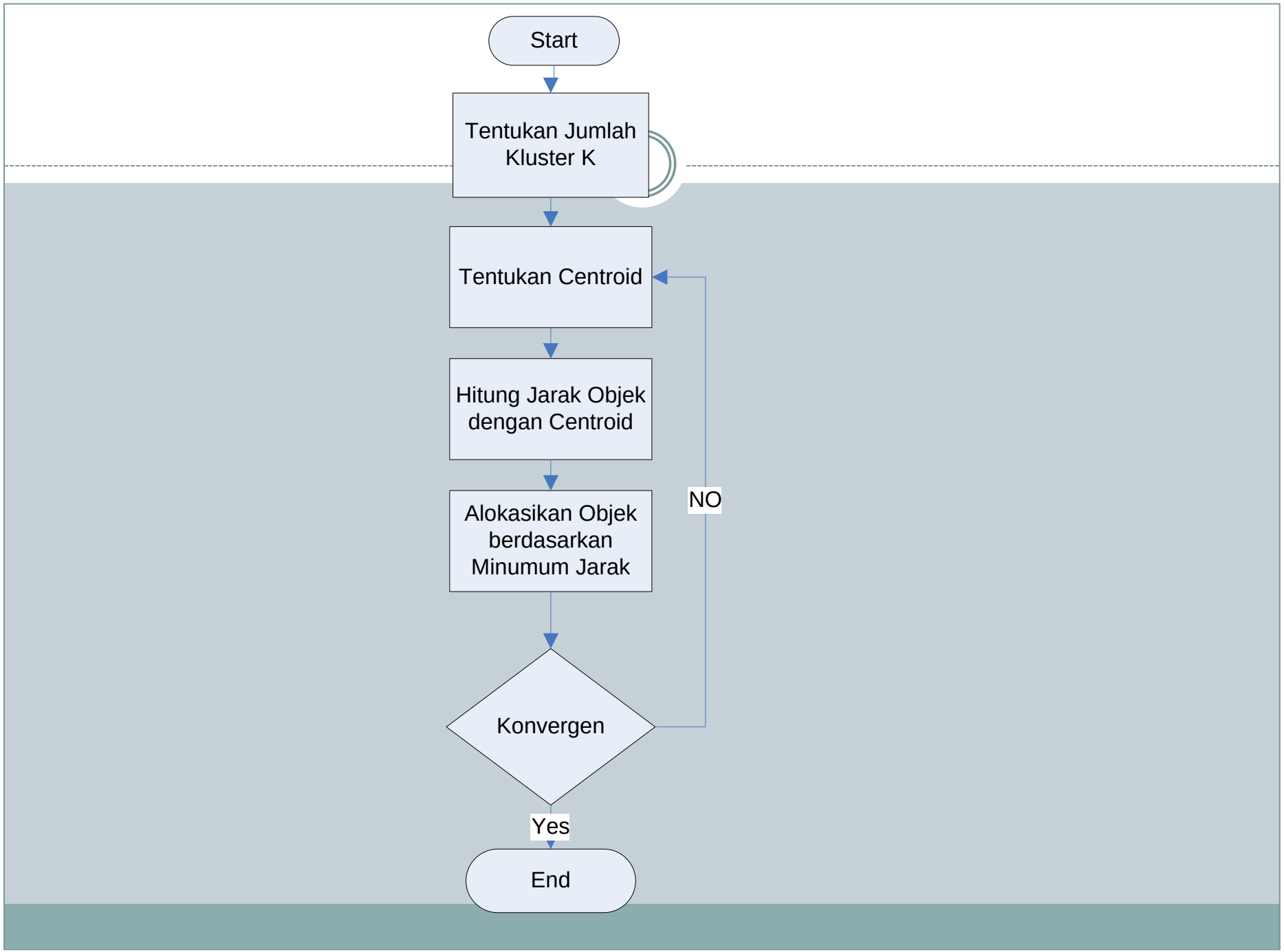
- *Pararel threshold* secara prinsip sama dengan sequential threshold hanya saja dilakukan dengan melakukan pemilihan terhadap beberapa objek awal *cluster* sekaligus dan kemudian melakukan penggabungan objek kedalamnya secara bersamaan.
- *Optimizing threshold* merupakan pengembangan dari sequential dan pararel dengan melakukan optimalisasi penempatan objek dengan melakukan *reassigned* ke dalam *cluster* untuk mengoptimalisasikan suatu kriteria secara menyeluruh, seperti *average within distance* untuk sejumlah *cluster* tertentu

Algoritma *K-Means Cluster Analysis*

- Jika diberikan sekumpulan data $X=(x_1,x_2,....x_n)$ maka algoritma *k-means cluster analysis* akan mempartisi X dalam *k* buah *cluster*, setiap *cluster* memiliki *centroid* (titik tengah) atau mean dari data-data dalam *cluster* tersebut.
- Pada tahap awal algoritma *k-means cluster analysis* akan memilih secara acak *k* buah data sebagai *centroid* (titik tengah), kemudian jarak antara data dengan *centroid* dihitung dengan menggunakan *Euclidean distance*, data akan ditempatkan dalam *cluster* yang terdekat dihitung dari titik tengah *cluster*.
- *Centroid* baru akan ditetapkan jika semua data sudah ditempatkan dalam *cluster* terdekat.
- Proses penentuan *centroid* dan penempatan data dalam *cluster* diulangi sampai nilai *centroid* konvergen (*centroid* dari semua *cluster* tidak berubah lagi)

- Secara umum *K-Means Cluster analysis* menggunakan algoritma sebagai berikut :

1. Tentukan *k* sebagai jumlah *cluster* yang akan di bentuk
2. Bangkitkan *k Centroid* (titik pusat *cluster*) awal secara *random*
3. Hitung jarak setiap data ke masing-masing *centroid* dari masing-masing *cluster*
4. Alokasikan masing-masing data ke dalam *centroid* yang paling terdekat
5. Lakukan iterasi, kemudian tentukan posisi *centroid* baru dengan cara menghitung rata-rata dari data-data yang berada pada *centroid* yang sama
6. Ulangi langkah 3 jika posisi *centroid* baru dan *centroid* lama tidak sama


Start
Tentukan Jumlah Kluster K
Tentukan Centroid
Hitung Jarak Objek dengan Centroid
Alokasikan Objek berdasarkan Minumum Jarak
Konvergen
NO
Yes
End

Menentukan Banyaknya *Cluster k*

- Untuk menentukan nilai banyaknya *cluster k* dilakukan dengan beberapa pertimbangan sebagai berikut :

1. Pertimbangan teoritis, konseptual, praktis yang mungkin diusulkan untuk menentukan berapa banyak jumlah *cluster*.
2. Besarnya relative *cluster* seharusnya bermanfaat, pemecahan *cluster* yang menghasilkan 1 objek anggota *cluster* dikatakan tidak bermanfaat sehingga hal ini perlu untuk dihindari.

Menentukan *Centroid*

- Penentuan *centroid* awal dilakukan secara *random*/acak dari data/objek yang tersedia sebanyak jumlah kluster *k*, kemudian untuk menghitung *centroid cluster* berikutnya ke *i*, v_i digunakan rumus sebagai berikut :

$$V_k = \frac{\sum_{i=1}^{N_i} X_i}{N_k}$$

- V_k : *centroid* pada *cluster* ke *k*
- *Xi* : Data ke *i*
- N_k : Banyaknya objek/jumlah data yang menjadi anggota *cluster* ke *k*

Menghitung Jarak Antara Data Dengan *Centroid*

- Untuk menghitung jarak antara data dengan *centroid* terdapat beberapa cara yang dapat dilakukan yaitu *Manhattan/City Block distance* (L_1), *Euclidean Distance* (L_2). Jarak antara dua titik X_1 dan X_2 pada *manhattan/citi block* dihitung dengan menggunakan rumus :

$$D_{L_1}(x_2, x_1) = \|x_2 - x_1\|_1 = \sum_{j=1}^{p} |x_{2j} - x_{1j}|$$

- Dimana P : Dimensi data
- | . | : Nilai Absolut

- Sedangkan untuk *euclidean distance* jarak antara data dengan *centroid* dihitung dengan menggunakan rumus :

$$D_{L_2}(x_2, x_1) = \|x_2 - x_1\|_2 = \sqrt{\sum_{j=1}^{p} (x_{2j} - x_{1j})^2}$$

- Dimana
- P : Dimensi data
- | . | : Nilai Absolut

Pengalokasian Ulang Data Kedalam Masing-masing *Cluster*

- Untuk melakukan pengalokasian data kedalam masing-masing *cluster* pada saat iterasi dilakukan secara umum dengan dua cara yaitu dengan cara pengalokasian dengan cara *hard k-means*, dimana secara tegas setiap objek dinyatakan sebagai anggota *cluster* satu dan tidak menjadi anggota *cluster* lainnya.
- Cara lain adalah dengan cara *fuzzy k-means* dimana masing-masing objek diberikan nilai kemungkinan untuk bisa bergabung dengan setiap *cluster* yang ada.

- *Hard K-means,* pengalokasian kembali objek kedalam masing-masing *cluster* pada metoda *hard K-means* didasarkan pada perbandingan jarak antara data dengan *centroid* setiap *cluster* yang ada, objek dialokasikan secara tegas kedalam *cluster* yang mempunyai jarak ke *centroid* terdekat dengan data tersebut. Pengalokasian ini dirumuskan sebagai berikut :

$$a_{ik} = \begin{cases} 1 & d = \min\{D(x_k, v_i)\} \\ 0 & lainnya \end{cases}$$

- a_{ik} : keanggotaan data atau objek ke *k* pada *cluster* ke i
- v_i : Nilai *centroid cluster* ke i

- *fuzzy k-means*, pada *fuzzy k-means* atau lebih sering disebut *fuzzy c-means* mengalokasikan kembali objek atau data kedalam masing-masing *cluster* dengan menggunakan *membership function*, u_{ik} ,yang merujuk pada seberapa besar suatu objek atau data bisa menjadi anggota suatu *cluster*.
- Pada *fuzzy k-means* yang diusulkan oleh Bezdek diperkenalkan juga suatu variable *m* yang merupakan *weighting exponent* dari *membership function*. *m* mempunyai wilayah nilai $m>1$, sampai sekarang belum jelas berapa nilai *m* yang optimal dalam melakukan proses optimalisasi suatu permasalahan *clustering*

- Nilai *m* yang umum digunakan adalah 2. *Membership function* untuk suatu data kedalam suatu *cluster* tertentu dihitung dengan menggunakan rumus :

$$u_{ik} = \sum_{j=1}^{c} \left(\frac{D(x_k, v_i)}{D(x_k, v_j)} \right)^{\frac{2}{m-1}}$$

- Dimana
- u_{ik} : membership function untuk data atau objek ke *k* pada *cluster* ke *i*
- v_i : Nilai *centroid cluster* ke *i*
- *m* : *Weighting component*
- c : Jumlah *cluster*

Konvergensi

- Pengecekan *konvergensi* dilakukan dengan membandingkan matrik *group assignment* pada iterasi sebelumnya dengan matrik *group assignment* pada iterasi yang sedang berjalan.
- Jika hasilnya sama maka *algoritma k-means cluster analysis* sudah *konvergen*, tetapi jika berbeda maka belum *konvergen* sehingga perlu dilakukan iterasi berikutnya

Menilai Kualitas *Cluster*

- Hasil dari *cluster analysis* yang bagus jika setiap *cluster* memiliki tingkat similaritas yang tinggi satu sama lain (*internal homogeneity*) diukur dengan variance dalam *cluster* V_w yang sama sekali berbeda dengan nilai anggota *cluster* yang lain (*external homogeneity*) yang diukur dengan varian antar *cluster* V_b
- *Cluster* dianggap ideal jika mempunya V_w seminimal mungkin dan V_b semaksimal mungkin, sehingga nilai *homogenity* dapat dirumuskan sebagai berikut :

$$V_{Min} = \frac{V_w}{V_b}$$

- untuk rumus ini maka semakin kecil nilai V_{min} maka *homogenity* semakin bagus,
- atau *homogenity* juga dapat dirumuskan sebagai berikut :

$$V_{Max} = \frac{V_b}{V_w}$$

- untuk rumus ini maka semakin besar nilai V_{max} maka *homogenity* semakin bagus

- Untuk menghitung nilai varians dari semua data pada tiap *cluster* dapat dilakukan dengan menggunakan rumus :

$$v_c^{\,2} = \frac{1}{n_c - 1} \sum_{i=1}^{n_c} \left(d_i - \overline{d_i} \right)^2$$

Dimana

V_c^2 =variance pada *cluster* c

c = 1..k dimana k = jumlah *cluster*

n_c = Jumlah data pada *cluster* ke c

d_i = data ke– *i* pada suatu *cluster*

$\overline{d_i}$ = rata-rata atau *centroid* dari data pada suatu *cluster*

- Sedangkan menghitung variance dalam *cluster* dapat dihitung dengan menggunakan rumus :

$$v_w = \frac{1}{N-k} \sum_{i=1}^{k} (n_i - 1) \cdot v_i^2$$

- Dimana V_w = Varians dalam *cluster*
- N = Jumlah semua data
- k = Banyaknya *cluster*
- n_i = Jumlah data dalam *cluster* ke i
- v_i^2 = Variance pada *cluster* ke i

- Sedangkan untuk menghitung varians antar *cluster* dihitung dengan menggunakan rumus :

$$v_b = \frac{1}{k-1} \sum_{i=1}^{k} n_i \left(\overline{d_i} - \overline{d} \right)^2$$

- Dimana

$$\overline{d} = rata - rata\ \overline{d}_i$$

- Sedangkan nilai variance dari semua *cluster* diperoleh dengan membagi nilai variance dalam *cluster* dengan nilai variance antar *cluster*, dimana semakin kecil nilai tersebut maka semakin bagus *cluster* yang dihasilkan.

Beberapa Permasalahan *K-Means Cluster Analysis*

- Terdapat beberapa permasalahan yang sering ditemukan pada pemakaian algoritma *K-means Cluster Analysis*, antara lain yaitu :

1. Pemilihan jumlah custer yang tepat
2. Ditemukannya beberapa hasil *cluster* yang berbeda.
3. Nilai distance yang sama, sehingga berpengaruh pada alokasi data dalam *cluster*
4. Kegagalan *Konvergensi*
5. Pendeteksian Outlier

Contoh Penerapan Algoritma *K-Means Cluster Analysis*

- Misalkan kita mempunyai dua variable X_1 dan X_2 dengan masing-masing mempunyai item-item A, B, C dan D sebagai berikut :

Item	Observasi	
	X_1	X_2
A	1	1
B	2	1
C	4	3
D	5	4

- Tujuannya adalah membagi semua item menjadi 2 *cluster* ($k = 2$) , dengan menggunakan algoritma yang disebutkan diatas maka langkah-langkah yang dikerjakan adalah sebagai berikut :

1. Tentukan k sebagai jumlah *cluster* yang akan di bentuk $k = 2$
2. Bangkitkan *k Centroid* (titik pusat *cluster*) awal secara *random* Secara *random* kita tentukan A dan B sebagai *centroid* yang pertama, sehingga diperoleh $c_1=(1,1)$ dan $c_2=(2,1)$
3. Hitung jarak setiap data ke masing-masing *centroid* dari masing-masing *cluster* dengan *Euclidian distance* sebagai berikut :

$$D_{L_2}(x_2, x_1) = \|x_2 - x_1\|_2 = \sqrt{\sum_{j=1}^{p} (x_{2j} - x_{1j})^2}$$

Dimana

- P : Dimensi data
- | . | : Nilai Absolut

$D(C_1,B) = \sqrt{(2-1)^2+(1-1)^2} = 1$

$D(C_1,C) = \sqrt{(4-1)^2+(3-1)^2} = 3,61$

$D(C_1,D) = \sqrt{(5-1)^2+(4-1)^2} = 5$

$D(C_2,A) = \sqrt{(1-2)^2+(1-1)^2} = 1$

$D(C_2,B) = \sqrt{(2-2)^2+(1-1)^2} = 0$

$D(C_2,C) = \sqrt{(4-2)^2+(3-1)^2} = 2,83$

$D(C_2,D) = \sqrt{(5-2)^2+(4-1)^2} = 4,24$

- Sehingga *distance* yang diperoleh adalah sebagai berikut

Cluster Centroid	Distance			
	A	B	C	D
C1	0	1	3,61	5
C2	1	0	2,83	4,24

4. Alokasikan masing-masing data ke dalam *centroid* yang paling terdekat

- Proses alokasi dilakukan dengan melihat minimum distance.
- Dari table distance diatas maka terlihat bahwa jarak item A terdekat pada *cluster* C1 sehingga item A dialokasikan kepada *cluster* C_1,
- sementara item B, Item C, Item D jarak terdekatnya pada *cluster* C_2, sehingga item B, C, D dialokasikan pada *cluster* C_2.

- Dengan menggunakan rumus alokasi dibawah ini,

$$a_{ik} = \begin{cases} 1 & d = \min\{D(x_k, v_i)\} \\ 0 & lainnya \end{cases}$$

- Maka diperoleh *table group assigmentnya* adalah sebagai berikut :

A	B	C	D
1	0	0	0
0	1	1	1

5. Lakukan iterasi-1, kemudian tentukan posisi *centroid* baru dengan cara menghitung rata-rata dari data-data yang berada pada *centroid* yang sama. Dengan menggunakan rumus,

$$V_i = \frac{\sum_{k=1}^{N_i} X_k}{N_i}$$

- Maka diperoleh *centroid* baru untuk kedua *cluster* tersebut adalah

C_1 = (1,1), karena beranggotakan 1 anggota

$$C_{2(x_1)} = \frac{2+4+5}{3} = 3,67$$

$$C_{2(x_2)} = \frac{1+3+4}{3} = 2,67$$

C_2=(3.67, 2.67)

6. Ulangi langkah 3 jika posisi *centroid* baru dan *centroid* lama tidak sama, karena nilai *centroid*nya berbeda maka langkah no 3 diulangi kembali sebagai berikut :

- $D^1(C_1,A) = \sqrt{(1-1)^2+(1-1)^2} = 0$
- $D^1(C_1,B) = \sqrt{(2-1)^2+(1-1)^2} = 1$
- $D^1(C_1,C) = \sqrt{(4-1)^2+(3-1)^2} = 3,61$
- $D^1(C_1,D) = \sqrt{(5-1)^2+(4-1)^2} = 5$
- $D^1(C_2,A) = \sqrt{(1-3,67)^2+(1-2,67)^2} = 3,14$
- $D^1(C_2,B) = \sqrt{(2-3,67)^2+(1-2,67)^2} = 2,36$
- $D^1(C_2,C) = \sqrt{(4-3,67)^2+(3-2,67)^2} = 0,47$
- $D^1(C_2,D) = \sqrt{(5-3,67)^2+(4-2,67)^2} = 1,89$

- Sehingga *distance* yang diperoleh pada iterasi 1 adalah sebagai berikut

Cluster Centroid	Distance			
	A	B	C	D
C1	0	1	3,61	5
C2	3,14	2,36	0,47	1,89

- Alokasikan masing-masing data ke dalam *centroid* yang paling terdekat
- Maka diperoleh *table group assigment*nya pada iterasi 1 adalah sebagai berikut :

A	B	C	D
1	1	0	0
0	0	1	1

- Karena hasil *table group assignment* pada iterasi 1 berbeda dengan *table group assignment* sebelumya maka hasilnya belum konvergen sehingga perlu dilakukan iterasi berikutnya, sebagai berikut
- Lakukan iterasi-2, tentukan posisi *centroid* baru dengan cara menghitung rata-rata dari data-data yang berada pada *centroid* yang sama.
- Maka diperoleh *centroid* baru untuk kedua *cluster* tersebut adalah

$$C_{1(x_1)} = \frac{1+2}{2} = 1,5$$

$$C_{1(x_2)} = \frac{1+1}{2} = 1$$

- $C_1=(1.5, 1)$

$$C_{2(x_1)} = \frac{4+5}{2} = 4,5$$

$$C_{2(x_2)} = \frac{3+4}{2} = 3,5$$

- $C_2=(4.5, 3.5)$

- karena nilai *centroid*-nya berbeda dengan iterasi 1 maka langkah berikutnya menghitung kembali *distance*-nya sebagai berikut :

- $D^2(C_1,A) = \sqrt{(1-1,5)^2+(1-1)^2} = 0,5$
- $D^2(C_1,B) = \sqrt{(2-1,5)^2+(1-1)^2} = 0,5$
- $D^2(C_1,C) = \sqrt{(4-1,5)^2+(3-1)^2} = 3,2$
- $D^1(C_1,D) = \sqrt{(5-1,5)^2+(4-1)^2} = 4,61$
- $D^2(C_2,A) = \sqrt{(1-4,5)^2+(1-3,5)^2} = 4,30$
- $D^2(C_2,B) = \sqrt{(2-4.5)^2+(1-3,5)^2} = 3,54$
- $D^2(C_2,C) = \sqrt{(4-4,5)^2+(3-3,5)^2} = 0,71$
- $D^2(C_2,D) = \sqrt{(5-4,5)^2+(4-3,5)^2} = 0,71$

- Sehingga *distance* yang diperoleh pada iterasi 1 adalah sebagai berikut

Cluster Centroid	Distance			
	A	B	C	D
C1	0,5	0,5	3,2	4,61
C2	4,3	3,54	0,71	0,71

- Alokasikan masing-masing data ke dalam *centroid* yang paling terdekat
- Maka diperoleh *table group assigmentnya* pada iterasi 2 adalah sebagai berikut :

A	B	C	D
1	1	0	0
0	0	1	1

- Dari hasil *table assignment* pada iterasi 2 ternyata hasilnya sama dengan *table group assignment* pada iterasi 1 sehingga pada iterasi 2 ini *sudah konvergen* sehingga tidak perlu dilakukan iterasi kembali, dan hasil akhir *cluster* yg diperoleh adalah :

Item	Observasi		Cluster
	X_1	X_2	
A	1	1	1
B	2	1	1
C	4	3	2
D	5	4	2