BAB 2

TINJAUAN PUSTAKA

2.1 Metode *Clustering*

Clustering adalah metode penganalisaan data, yang sering dimasukkan sebagai salah satu metode *Data Mining*, yang tujuannya adalah untuk mengelompokkan data dengan karakteristik yang sama ke suatu ‘wilayah’ yang sama dan data dengan karakteristik yang berbeda ke ‘wilayah’ yang lain.

Ada beberapa pendekatan yang digunakan dalam mengembangkan metode *clustering*. Dua pendekatan utama adalah *clustering* dengan pendekatan partisi dan *clustering* dengan pendekatan hirarki (Oliveira *et al*, 2007). *Clustering* dengan pendekatan partisi atau sering disebut dengan *partition-based clustering* mengelompokkan data dengan memilah-milah data yang dianalisa ke dalam cluster-cluster yang ada. *Clustering* dengan pendekatan hirarki atau sering disebut dengan *hierarchical clustering* mengelompokkan data dengan membuat suatu hirarki berupa dendogram dimana data yang mirip akan ditempatkan pada hirarki yang berdekatan dan yang tidak pada hirarki yang berjauhan. Di samping kedua pendekatan tersebut, ada juga *clustering* dengan pendekatan *automatic mapping* (Self-Organising Map/SOM).

2.2 *Clustering* Dengan Pendekatan Partisi

2.2.1 *K-Means*

Salah satu metode yang banyak digunakan dalam melakukan *clustering* dengan partisi ini adalah metode k-means.

Secara umum metode k-means ini melakukan proses pengelompokan dengan prosedur sebagai berikut (Maimon *et al*, 2010):

1. Tentukan jumlah cluster
2. Alokasikan data secara random ke cluster yang ada
3. Hitung rata-rata setiap cluster dari data yang tergabung di dalamnya
4. Alokasikan kembali semua data ke cluster terdekat
5. Ulang proses nomor 3, sampai tidak ada perubahan atau perubahan yang terjadi masih sudah di bawah treshold

Prosedur dasar ini bisa berubah mengikuti pendekatan pengalokasian data yang diterapkan, apakah *crisp* atau *fuzzy*. Setelah meneliti *clustering* dari sudut yang lain, ditemukan bahwa *k-means clustering* mempunyai beberapa kelemahan.

2.2.2 *Mixture Modelling (Mixture Modeling)*

Mixture modelling (mixture modeling) merupakan metode pengelompokan data yang mirip dengan k-means dengan kelebihan penggunaan distribusi statistik dalam mendefinisikan setiap cluster yang ditemukan. Dibandingkan dengan k-means yang hanya menggunakan cluster center, penggunaan distribusi statistik ini mengijinkan kita untuk (Hastie *et al*, 2010):

1. Memodel data yang kita miliki dengan setting karakteristik yang berbeda-beda
2. Jumlah cluster yang sesuai dengan keadaan data bisa ditemukan seiring dengan proses pemodelan karakteristik dari masing-masing cluster
3. Hasil pemodelan *clustering* yang dilaksanakan bisa diuji tingkat keakuratannya

Distribusi statistik yang digunakan bisa bermacam-macam mulai dari yang digunakan untuk data categorical sampai yang continuous, termasuk di antaranya distribusi binomial, multinomial, normal dan lain-lain. Beberapa distribusi yang bersifat tidak normal seperti distribusi Poisson, von-Mises, Gamma dan Student t, juga

diterapkan untuk bisa mengakomodasi berbagai keadaan data yang ada di lapangan. Beberapa pendekatan multivariate juga banyak diterapkan untuk memperhitungkan tingkat keterkaitan antara variabel data yang satu dengan yang lainnya.

2.3 *Clustering* dengan Pendekatan Hirarki

Clustering dengan pendekatan hirarki mengelompokkan data yang mirip dalam hirarki yang sama dan yang tidak mirip di hirarki yang agak jauh. Ada dua metode yang sering diterapkan yaitu *agglomerative hieararchical clustering* dan *divisive hierarchical clustering*. Agglomerative melakukan proses *clustering* dari *N* cluster menjadi satu kesatuan cluster, dimana *N* adalah jumlah data, sedangkan divisive melakukan proses *clustering* yang sebaliknya yaitu dari satu cluster menjadi *N* cluster.

Beberapa metode *hierarchical clustering* yang sering digunakan dibedakan menurut cara mereka untuk menghitung tingkat kemiripan. Ada yang menggunakan *Single Linkage*, *Complete Linkage*, *Average Linkage*, *Average Group Linkage* dan lain-lainnya. Seperti juga halnya dengan *partition-based clustering*, kita juga bisa memilih jenis jarak yang digunakan untuk menghitung tingkat kemiripan antar data.

Salah satu cara untuk mempermudah pengembangan dendogram untuk hierarchical *clustering* adalah dengan membuat similarity matrix yang memuat tingkat kemiripan antar data yang dikelompokkan. Tingkat kemiripan bisa dihitung dengan berbagai macam cara seperti dengan Euclidean Distance Space. Berangkat dari similarity matrix ini, kita bisa memilih lingkage jenis mana yang akan digunakan untuk mengelompokkan data yang dianalisa (Everitt *et al*, 2011).

2.3.1 Agglomerative *Clustering*

Didalam *agglomerative clustering* dimulai dengan mewakili setiap data observasi dengan memasukkannya sebagai cluster tunggal. Kemudian mencari pasangan cluster yang berbeda. Pasangan cluster tersebut disatukan sehingga menjadi satu cluster yang kemudian pada langkah-langkah selanjutnya akan menghasilkan pengurangan jumlah cluster dan akhirnya menghasilkan hanya satu cluster yang mewakili semua data.

Untuk itu ukuran untuk menentukan ketidaksamaan setiap cluster harus ditentukan terlebih dahulu (Everitt *et al*, 2011).

2.3.1.1 Single Linkage

Metode *Single Linkage clustering(SL)* sering juga disebut dengan *nearest-neighbor tehnique* dimana pencarian pasangan jarak untuk disatukan berdasarkan pengukuran jarak terdekat. Sebut saja *G* dan *H* adalah dua cluster yang akan disatukan. Ketidaksamaan jarak *d(G,H)* akan di hitung lalu dengan cara membandingkan setiap jarak anggota kelompok dari G_i terhadap jarak setiap anggota kelompok dari $H_{i'}$ kemudian mencari pasangan yang jaraknya terdekat.

$$d_{SL}(G,H) = min(d_{ii'}); i \in G; i' \in H \quad \text{...(2.1) (Hastie } et\ al\text{, 2010)}$$

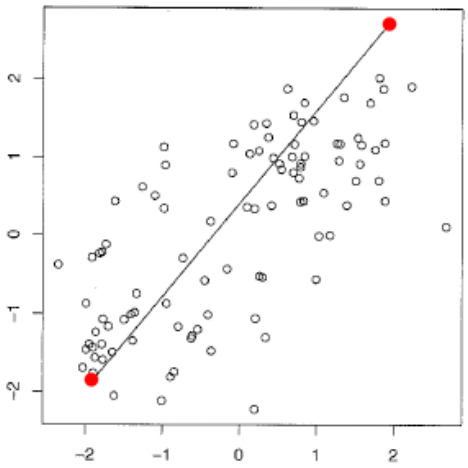
2.3.1.2 Complete Linkage

Pada metode *Complete Linkage Agglomerative Clustering* (CL) biasa disebut dengan metode *furthest neighbor technique*. Metode ini secara umum prosesnya hampir sama dengan metode single linkage tetapi pada pencarian pasangan, metode complete linkage mencari pasangan yang jaraknya terjauh dari nilai observasi.

$$d_{CL}(G,H) = max(d_{ii'}); i \in G; i' \in H \quad \text{...(2.2) (Hastie } et\ al\text{, 2010)}$$

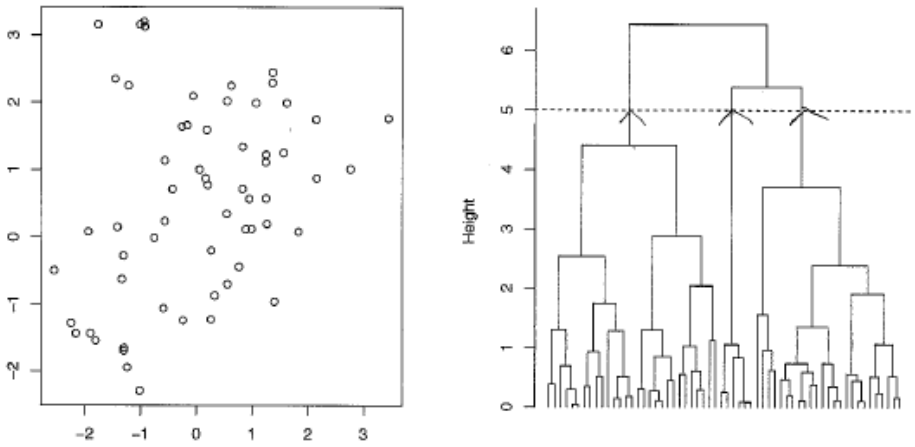
Metode ini didasarkan pada jarak maksimum. Pada metode ini juga mengelompokkan data pada jarak yang terjauh terlebih dahulu. Metode ini dikenal dengan nama tetangga terjauh. Sesuai dengan persamaan (2.2).

Ketidak-samaan antara *G*, *H* adalah ketidak samaan antara dua titik pada kelompok yang bertentangan. Ketidak samaan d_{ij} adalah jarak yang ditandai dengan warna dari kedua titik pada gambar dibawah ini.



Gambar 2.1 Jarak dua titik yang terjauh pada algoritma Agglomerative *Clustering* Complete Linkage.

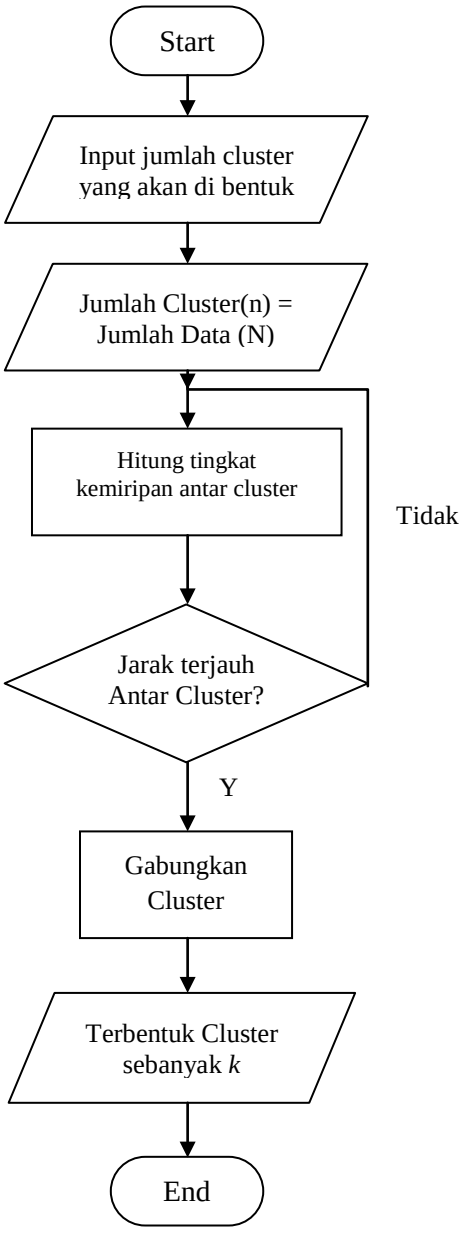
Hasil dari Algoritma Complete Linkage dibuat dalam sebuah dendogram yang biasa disebut diagram pohon. Setiap cabang akan bertemu dan disatukan. Selanjutnya proses ini akan memotong cabang tree dan kemudian $d_{[CL]}$ akan menghasilkan titik terjauh. Pada Gambar 2.2 tree akan di potong pada h=5.



Gambar 2.2 Contoh Pemotongan cabang dendogram pada h=5

Hasil Algoritma Fuzzy C-Means yang menggunakan Complete Linkage sebagai algoritma yang menentukan titik pusat cluster akan menghasilkan nilai fungsi objektif objektif yang berbeda dan nilai tersebut akan di bandingkan dengan menggunakan grafik oleh Algoritma Fuzzy C-Means biasa dengan juga memperhitungkan jumlah perulangan yang didapat dan besar iterasi.

Dengan bobot dan parameter serta data yang sama di harapkan algoritma fuzzy c-means yang dikembangkan menghasilkan tingkat efisiensi dari segi waktu yang paling utama adalah jumlah iterasi atau perulangan untuk mencapai $P_t - P_{t-1} < \xi$ lebih baik dari sebelumnya.



Gambar 2.3 Flowchart algoritma Agglomerative *Clustering* Complete Linkage.

2.3.1.3 Pembuatan Centroid Data

Pembuatan centroid data atau pusat data didasari pada paper *multistage random sampling FCM Algorithm* yang menyatakan bahwa sekelompok kecil vector dapat digunakan untuk mengaproksimasi pusat cluster keseluruhan sekelompok besar data (Cheng *et al*, 1998).

Untuk itu pemilihan algoritma complete linkage yang mencari pusat cluster berdasarkan pasangan terjauh diharapkan tepat untuk memprediksi nilai pusat pusat cluster yang diteliti. Namun demikian pada algoritma complete linkage yang memilih pusat cluster dengan perbandingan maximum jarak *A* ke *B* akan mengakibatkan pusat cluster tersebut tetap condong pada jarak yang paling maximum sehingga pusat cluster tidak tepat untuk mewakili sekelompok nilai.

Pada pemodelan pencarian pusat cluster menggunakan algoritma complete linkage diubah menjadi nilai tengah dari perbandingan dua jarak minimum dan maximum.

$$V_{ij}(A,B) = max\big(d(A,B)\big) - \frac{1}{2}|\max(d(A,B)) - \min(d(A,B))| \ldots. (2.3)$$

Sehingga perbandingan pusat cluster dengan nilai tengah terdapat pada gambar berikut ini:

Gambar 2.4 Perbandingan pencarian pusat cluster, kiri Complete Linkage dan kanan Persamaan (2.3).

Sedangkan untuk perhitungan jarak untuk pencarian fungsi keanggotaan baru pada algoritma C-Means ketika setelah melakukan proses inisialisasi titik awal.

$$d_i = \sum_{j=1}^{c}\sum_{k=1}^{s}\left(X_{ik} - V_{jk}\right)^2 \ldots. (2.4)$$

2.3.1.4 Average Linkage

Ukuran yang menjadi tolak ukur ketidaksamaan untuk menyatukan kedua cluster tidak hanya berdasarkan kedekatan jarak maupun bedasarkan jarak terjauh. Pada metode lain terdapat metode Average Linkage atau disebut juga *Group Average*(GA) yang mencari pasangan dengan melihat rata-rata jarak setiap nilai observasinya.

$$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{i' \in H} d_{ii'}$$ .....(2.5) (Hastie *et al*, 2010)

2.3.2 Divisive *Clustering*

Algoritma ini membagi satu cluster yang berisi banyak data menjadi beberapa cluster kecil. Divisive *clustering* memulai dengan memasukkan semua data kedalam satu cluster tunggal lalu membagi cluster yang ada menjadi dua anak-anak cluster hingga secara rekursif membagi menjadi *N* buah cluster untuk setiap nilai observasi.

Sebagai pengukuran untuk melihat ketidaksamaan untuk setiap cluster adalah:

$$\bar{d}_G = \frac{1}{N_G} \sum_{i \in G} \sum_{i' \in H} d_{ii'}$$ ….(2.6) (Hastie *et al*, 2010)

2.4 *Clustering* Dengan Pendekatan Automatic Mapping

2.4.1 *Self-Organising Map (SOM)*

Self-Organising Map (SOM) merupakan suatu tipe Artificial Neural Networks yang di-training secara unsupervised. SOM menghasilkan map yang terdiri dari output dalam dimensi yang rendah (2 atau 3 dimensi). Map ini berusaha mencari property dari input data. Komposisi input dan output dalam SOM mirip dengan komposisi dari proses feature scaling (multidimensional scaling).

Walaupun proses learning yang dilakukan mirip dengan Artificial Neural Networks, tetapi proses untuk meng-assign input data ke map, lebih mirip dengan K-Means dan

KNN Algorithm. Adapun prosedur yang ditempuh dalam melakukan *clustering* dengan SOM adalah sebagai berikut (Hastie *et al*, 2010):

1. Tentukan weight dari input data secara random
2. Pilih salah satu input data
3. Hitung tingkat kesamaan (dengan Eucledian) antara input data dan weight dari input data tersebut dan pilih input data yang memiliki kesamaan dengan weight yang ada (data ini disebut dengan Best Matching Unit (BMU))
4. Perbaharui weight dari input data dengan mendekatkan weight tersebut ke BMU dengan rumus:

   $Wv(t+1)=Wv(t) + Theta(v, t) \times Alpha(t) \times (D(t) – Wv(t))$…(2.7) (Hastie *et al*, 2010)

   Dimana:

   - *Wv(t)*: Weight pada saat ke-t
   - *Theta (v, t)*: Fungsi neighbourhood yang tergantung pada Lattice distance antara BMU dengan neuron v. Umumnya bernilai 1 untuk neuron yang cukup dekat dengan BMU, dan 0 untuk yang sebaliknya. Penggunaan fungsi Gaussian juga memungkinkan.
   - *Alpha (t)*: Learning Coefficient yang berkurang secara monotonic
   - *D(t)*: Input data
5. Tambah nilai t, sampai t < *Lambda*, dimana *Lambda* adalah jumlah iterasi

2.5 *Clustering* Dengan Pendekatan Berbasis Fuzzy

2.5.1 Fuzzy *Clustering* Means (Fuzzy C-Means)

Fuzzy *clustering* adalah proses menentukan derajat keanggotaan, dan kemudian menggunakannya dengan memasukkannya kedalam elemen data kedalam satu kelompok cluster atau lebih.

Hal ini akan memberikan informasi kesamaan dari setiap objek. Satu dari sekian banyaknya algoritma fuzzy *clustering* yang digunakan adalah algoritma fuzzy *clustering* c means. Vektor dari fuzzy *clustering*, $V=\{v_1, v_2, v_3,\ldots, v_c\}$, merupakan sebuah fungsi objektif yang di defenisikan dengan derajat keanggotaan dari data X_j dan pusat cluster V_j.

Algoritma fuzzy *clustering* c means membagi data yang tersedia dari setiap elemen data berhingga lalu memasukkannya kedalam bagian dari koleksi cluster yang dipengaruhi oleh beberapa kriteria yang diberikan. Berikan satu kumpulan data berhingga. $X= \{x_1,\ldots, x_n\}$ dan pusat data.

$$J_m(X,U,V) = \sum_{j=1}^{n}\sum_{i=1}^{c}\left(\mu_{ij}\right)^m d^2(X_j,V_i) \quad \ldots. (2.8)$$ (Valarmathie *et al*, 2009)

Dimana μ_{ij} adalah derajat keanggotaan dari X_j dan pusat cluster adalah sebuah bagian dari keanggotaan matriks $[\mu_{ij}]$. d^2 adalah akar dari *Euclidean distance* dan m adalah parameter fuzzy yang rata-rata derajat kekaburan dari setiap data derajat keanggotaan tidak lebih besar dari 1,0 (Valarmathie *et al*, 2009)

Output dari *Fuzzy C-Means* merupakan deretan pusat *cluster* dan beberapa derajat keanggotaan untuk tiap-tiap titik data. Informasi ini dapat digunakan untuk membangun suatu *fuzzy inference system*.

2.5.2 Langkah Algoritma *Fuzzy Clustering Means* (FCM)

Algoritma Fuzzy C-Means adalah sebagai berikut:

1. Input data yang akan dicluster *X*, berupa matriks berukuran *n x m* (*n*=jumlah sample data, *m*=atribut setiap data). X_{ij}=data sample ke-*i* (*i*=1,2,…,*n*), atribut ke-*j* (*j*=1,2,…,*m*).

2. Tentukan :

    1. Jumlah cluster = *c*
    2. Pangkat = *w*
    3. Maksimum iterasi = *MaxIter*
    4. Error terkecil yang diharapkan = ξ
    5. Fungsi obyektif awal = *Po* = 0
    6. Iterasi awal = *t* = 0

3. Bangkitkan nilai acak μik, i=1,2,…,n; k=1,2,…,c sebagai elemen-elemen matriks partisi awal μik. μik adalah derajat keanggotaan yang merujuk pada seberapa besar kemungkinan suatu data bisa menjadi anggota ke dalam suatu cluster. Posisi dan nilai matriks dibangun secara random. Dimana nilai keangotaan terletak pada interval 0 sampai dengan 1. Pada posisi awal matriks partisi U masih belum akurat begitu juga pusat clusternya. Sehingga kecendrungan data untuk masuk suatu cluster juga belum akurat.

$$Q_i = \sum_{k=1}^{c} \mu_{ik} \quad \text{.....(2.9) (Bezdek, 1981)}$$

Langkah selanjutnya lakukan normalisasi data dengan menggunakan persamaan berikut:

$$\mu_{ik} = \frac{\mu_{ik}}{Q_i} \quad \text{……(2.10) (Bezdek, 1981)}$$

4. Hitung pusat *Cluster* ke-k: V_{kj} ,dengan *k*=1,2,…c dan *j*=1,2,…m. dimana X_{ij}adalah variabel fuzzy yang digunakan dan *w* adalah bobot.

$$V_{kj} = \frac{\sum_{i=1}^{n}\left(\left(\mu_{ik}\right)^{w} * X_{ij}\right)}{\sum_{i=1}^{n}\left(\mu_{ik}\right)^{w}} \quad \text{..... (2.11) (Das, 2013)}$$

5. Fungsi objektif digunakan sebagai syarat perulangan untuk mendapatkan pusat cluster yang tepat. Sehingga diperoleh kecendrungan data untuk masuk ke *cluster* mana pada *step* akhir.

6. Hitung fungsi obyektif pada iterasi ke-*t*, P_t

$$P_t = \sum_{i=1}^{n}\sum_{k=1}^{c}\left(\left[\sum_{j=1}^{m}\left(X_{ij} - V_{kj}\right)^2\right]\left(\mu_{ik}\right)^{w}\right) \text{... (2.12) (Bezdek, 1981)}$$

7. Perhitungan fungsi objektif P_t dimana nilai variabel fuzzy X_{ij} di kurang dengan dengan pusat cluster V_{kj} kemudian hasil pengurangannya di kuadradkan lalu masing-masing hasil kuadrad di jumlahkan untuk dikali dengan kuadrad dari derajat keanggotaan μ_{ik} untuk tiap *cluster*. Setelah itu jumlahkan semua nilai di semua *cluster* untuk mendapatkan fungsi objektif P_t.

8. Hitung perubahan matriks partisi:

$$\mu_{ik} = \frac{\left[\sum_{j=1}^{m}\left(X_{ij} - V_{kj}\right)^2\right]^{\frac{-1}{w-1}}}{\sum_{k=1}^{c}\left[\sum_{j=1}^{m}\left(X_{ij} - V_{kj}\right)^2\right]^{\frac{-1}{w-1}}} \quad \text{……. (2.13) (Bezdek, 1981)}$$

Atau

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}} \; untuk\; d_{ik} = \|X_k - V_i\| > 0, \forall\, i\; dan\; k \text{ ... (2.14) (Bezdek, 1981)}$$

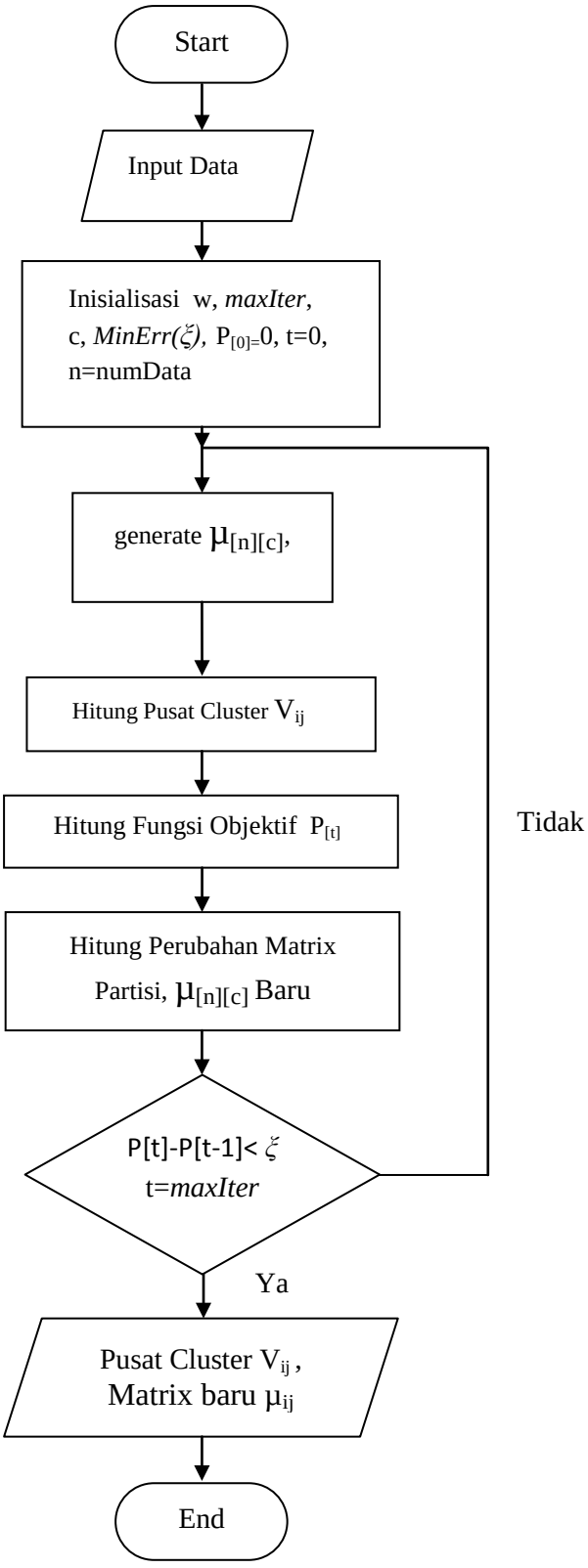
9. Dengan: i=1,2,…n dan k=1,2,..c. Untuk mencari perubahan matrik partisi μ_{ik},pengurangan nilai variabel fuzzy X_{ij} di lakukan kembali terhadap pusat cluster V_{kj}lalu dikuadradkan. Kemudian dijumlahkan lalu dipangkatkan dengan -1/(w-1) dengan bobot, w=2 hasilnya setiap data dipangkatkan dengan -1. Setelah proses perhitungan dilakukan, normalisasikan semua data derajat keanggotaan baru dengan cara menjumlahkan derajat keanggotaan baru k=1,…c, hasilnya kemudian dibagi dengan derajat keanggotaan yang baru. Proses ini dilakukan agar derajat keanggotaan yang baru mempunyai rentang antara 0 dan tidak lebih dari 1.

10. Cek kondisi berhenti, jika:($|Pt - Pt\text{-}1| < \xi$) atau (t>maxIter) maka berhenti, jika tidak, t=t+1, ulangi langkah ke-4.

11. Harapan yang di inginkan adalah sesuai persamaan, dimana

$$\sum_{j=1}^{c} u_{ik} = 1, 1 \leq i \leq n \ldots\ldots (2.15)$$(Bezdek, 1981)

$$u_{ik} \geq 0, 1 \leq i \leq n, 1 \leq j \leq c \ldots\ldots (2.16)$$



Gambar 2.5 Flowchart Fuzzy C-Means

2.6 Cluster Analysis

Dalam cluster analisis pengelompokan objek dilakukan berdasarkan kesamaan dan ketidaksamaan. Setiap objek yang tergabung didalam satu kelompok atau lebih dalam Fuzzy c-Means memiliki tingkat homogenitas yang tinggi dibandingkan objek lainnya. Untuk itu pengujian dapat dilakukan dengan melihat nilai variansi atau sebaran data. Variansi cluster dapat ditentukan dengan persamaan.

$$V_c^2 = \frac{1}{n_c - 1}\sum_{i=1}^{n_c}(x_i - \bar{x_c})^2 \ldots . (2.17)\ (weis, 2006)$$

Berdasarkan persamaan 2.17 yang menghasilkan variansi setiap cluster, maka kepadatan suatu cluster bisa didapat dengan analisis *variance within cluster*, sesuai dengan persamaan 2.18.

$$V_w = \frac{1}{N - c}\sum_{i=1}^{c}(n_i - 1).V_i^2 \ldots . (2.18)(weis, 2006)$$

Analisis yang lain adalah untuk melihat sebaran data antara cluster(*variance between cluster*) bisa dihitung dengan persamaan 2.19 dibawah ini.

$$V_b = \frac{1}{k - 1}\sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x}) \ldots . (2.19)(weis, 2006)$$

Cluster dengan nilai V_w minimum dapat merepresentasikan *Internal Homogenity* sehingga cluster tersebut lebih mendekati ideal. Sedangkan V_b dengan nilai terbesat memaparkan *External Homogenity*. Pada persamaan selanjutnya dapat menyatakan batasan variansi.

$$V = \frac{V_w}{V_b} \ldots . (2.20)(weis, 2006)$$