BAB 2

LANDASAN TEORI

2.1 *Clustering* dalam *Data Mining*

Data mining merupakan kegiatan eksplorasi dan analisis data dalam jumlah besar dengan tujuan untuk menemukan pola yang bermakna. Data yang akan diproses dalam *data mining* ada kalanya belum diketahui label kelasnya. Dalam hal pengelompokkan, data semacam ini dikelompokkan berdasarkan kemiripan dan ketidakmiripan antar datanya atau yang sering disebut dengan pembelajaran tidak terbimbing (*unsupervised learning*). Karena tidak ada label kelas yang digunakan dalam prosesnya, *clustering* sangat cocok digunakan untuk pengelompokkan data yang label kelasnya memang sulit didapatkan pada saat pembangkitan fitur. Label kelas baru dapat diberikan setelah proses *clustering* selesai dengan cara mengamati hasil *cluster* yang didapat. Teknik *data mining clustering* juga dapat digunakan untuk mengeksplorasi dan mengkarakteristikkan set data pada pembelajaran terbimbing (*supervised learning*) (Anil dan Richard, 1998).

Hasil *clustering* dikatakan baik jika data dalam satu *cluster* memiliki kemiripan yang tinggi dan memiliki ketidakmiripan yang tinggi dengan data di *cluster* lain. Hal yang perlu diperhatikan dalam *clustering* adalah bahwa kemiripan (*similarity*) harus didefinisikan berdasarkan pada atribut objek. Kualitas hasil *clustering* yang terbentuk bergantung pada data dan algoritma *clustering* yang digunakan. Algoritma *clustering* yang berbeda akan cocok dengan

jenis *dataset* yang berbeda pula. Oleh karena itu, diperlukan adanya kecocokan antara algoritma dan *dataset* yang akan diproses sehingga dapatdihasilkan hasil *clustering* yang baik.

Dalam kasus di dunia nyata, *clustering* digunakan salah satunya untuk melakukan segmentasi pasar atau pelanggan yang dapat membantu dalam *Customer Relationship Management* (CRM) (Anil dan Richard, 1998). Melalui segmentasi pelanggan, perusahaan dapat mengetahui kelompok-kelompok pelanggannya serta bagaimana karakteristik setiap kelompok tersebut, sehingga perusahaan dapat menerapkan strategi pemasaran yang sesuai untuk setiap kelompok pelanggan. Dilihat dari hal tersebut, dapat dikatakan bahwa hasil *clustering* akan sangat mempengaruhi proses pengambilan keputusan. Sehingga diperlukan adanya algoritma yang tepat untuk menghasilkan hasil *clustering* yang baik.

2.2 Data kategori

Ada beberapa tipe data yang biasa digunakan untuk mempresentasikan data di dalam *databases* dan *data analysis* (Z Huang, 1998). Tipe data yang biasanya digunakan dalam *relational databases* antara lain: *integer, float, double, character* dan *strings,* sedangkan tipe data (tipe variabel) dalam *data analysis* antara lain: *interval, ratio, binary, ordinal, nominal,* dsb.

Dalam algoritma *clustering* yang didiskusikan selanjutnya, hanya mempertimbangkan 2 tipe data yang umum yaitu *numeric* (numerik) dan

catergorical (kategori), kemudian mengasumsikan tipe lainnya bisa di petakan (*mapping*) ke dalam satu diantara kedua tipe data tersebut.

Banyak data di dunia nyata yang bertipe kategori, contohnya: *gender*, profesi, jabatan, dan kegemaran. Setiap atribut kategori memiliki sebuah himpunan nilai kategori seperti : {female, male} pada atribut *gender*. Data kategori adalah data yang menggambarkan objek yang hanya memiliki atribut kategori. Tidak seperti data numerik, data kategori memiliki nilai sendiri dan tidak memiliki urutan.

2.3 Teknik-teknik dalam *Clustering*

2.3.1 Algoritma *Weighted Dissimilarity Measure K-Modes* (WDM *K-Modes*)

Algoritma *K-Modes* adalah salah satu algoritma *clustering* yang digunakan untuk tipe data kategori. Data kategori merupakan data yang diambil dari suatu himpunan nilai tertentu, yang tidak harus berupa angka atau pecahan misalkan warna mata, warna kulit, dan nama negara. Data kategori yang mengandung nilai seperti [*male,female*], [*low,medium,high*] tidak dapat diukur secara geometri.

Misalkan X dan Y adalah dua data dengan fitur bertipe kategori. Ukuran ketidakmiripan di antara X dan Y dapat diukur dengan jumlah ketidakcocokan nilai dari fitur yang berkorespondensi dari dua data. Semakin kecil nilai ketidakcocokan, maka semakin mirip dua data tersebut. Metrik seperti ini sering disebut dengan *simple matching* (pencocokan sederhana).

Ukuran ketidaksamaan (*dissimilarity measure*) adalah ukuran yang digunakan untuk menghitung jarak antar objek (Z Huang, 1998). Pada langkah

ketiga algoritma *K-Modes* tradisional ukuran jarak ketidaksamaan (*dissimilarity measure*) yang digunakan adalah *Simple Matching Measure*. *Simple Matching Measure* dapat didefinisikan sebagai jumlah atribut yang tidak sama antara dua objek. Misalkan suatu objek t_1 = [x,y,y] dan t_2 = [x,y,a], jarak $t_{13} \neq t_{23}$ berdasarkan *hamming distance* maka diberi nilai 1 dan bernilai 0 karena jarak $t_{11} = t_{21}$. Jadi pada contoh, jumlah jarak t_1 dan t_2 adalah 0+0+1 = 1. Pada teknik ini tidak mempertimbangkan hubungan kesamaan yang ada pada nilai atribut sehingga menghasilkan kesamaan *intra cluster* yang lemah. Ketidakcocokan (*mismatching*) sederhana atau biasa disebut *dissimilarity* tidak menggunakan hubungan yang inheren antar atribut.

Pada *k-modes* menggunakan *weighted dissimilarity measure* jarak antar dua objek jika atribut $a_1 = a_2$ tidak lagi bernilai 0 tetapi 1-w_{ij}. Nilai w_{ij} adalah perkalian perbandingan nilai atribut di *cluster* dengan perbandingan nilai atribut di dataset seperti pada persamaan (Aranganayagi dan Thangevel, 2009). Hal ini membuat dalam menghasilkan pembentukan *cluster*-nya lebih rinci lagi, sehingga kesamaan *intra cluster* bertambah kuat. Pada iterasi pertama *weighted dissimilarity measure* masih menggunakan cara kerja perhitungan jarak pada *k-modes* tradisional karena iterasi pertama belum ada *cluster* yang terbentuk.

$$\delta(x_j, y_j) = \begin{cases} 1 - w_{i,j} & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases} \tag{2.1}$$

$$w_{ij} = \left(\frac{f\frac{v_{ij}}{c_i}}{|c_i|}\right) * \left(\frac{f\frac{v_{ij}}{D}}{|D|}\right) \tag{2.2}$$

Keterangan:

$f(v_{ij}/C_i)$ adalah jumlah objek di *cluster* C_i dengan nilai atribut v_{ij}

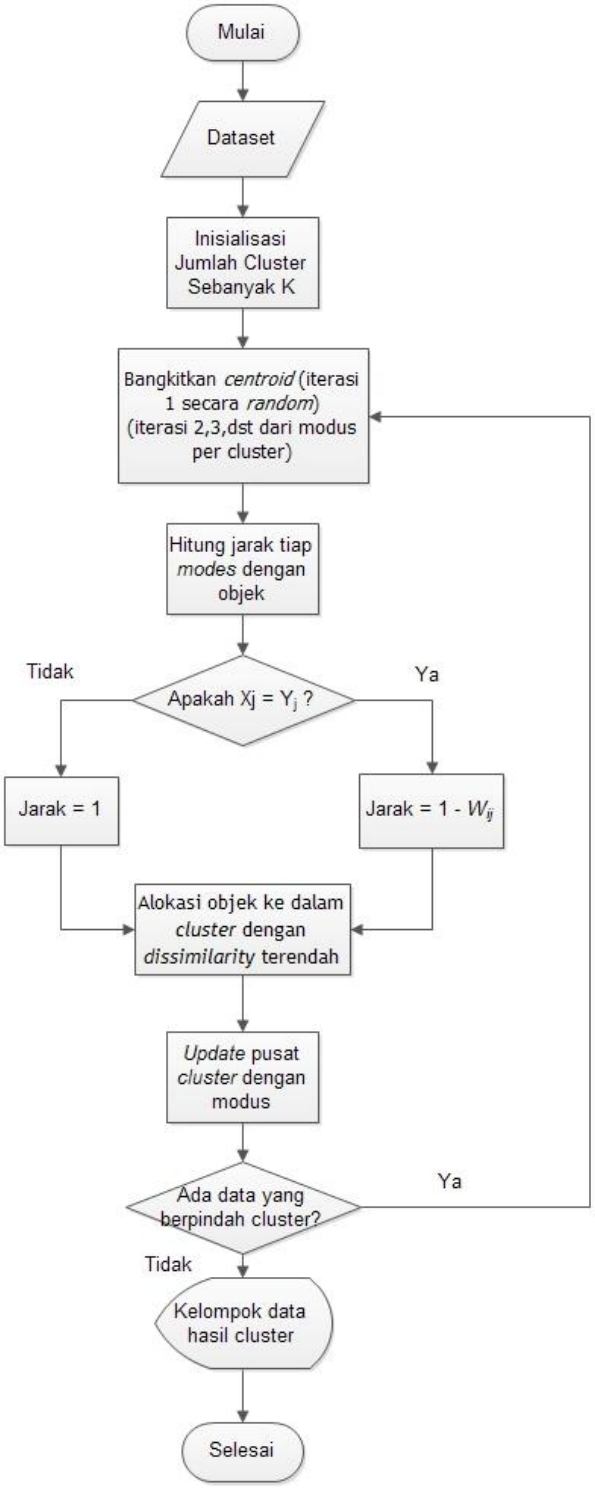
$f(v_{ij}/D)$ adalah jumlah objek di *dataset* dengan nilai atribut v_{ij}

C_i adalah jumlah objek di *cluster* C_i

D adalah jumlah objek di *dataset*

Langkah-langkah algoritma WDM K-Modes adalah sebagai berikut dan dapat dilihat di Gambar 2.1.

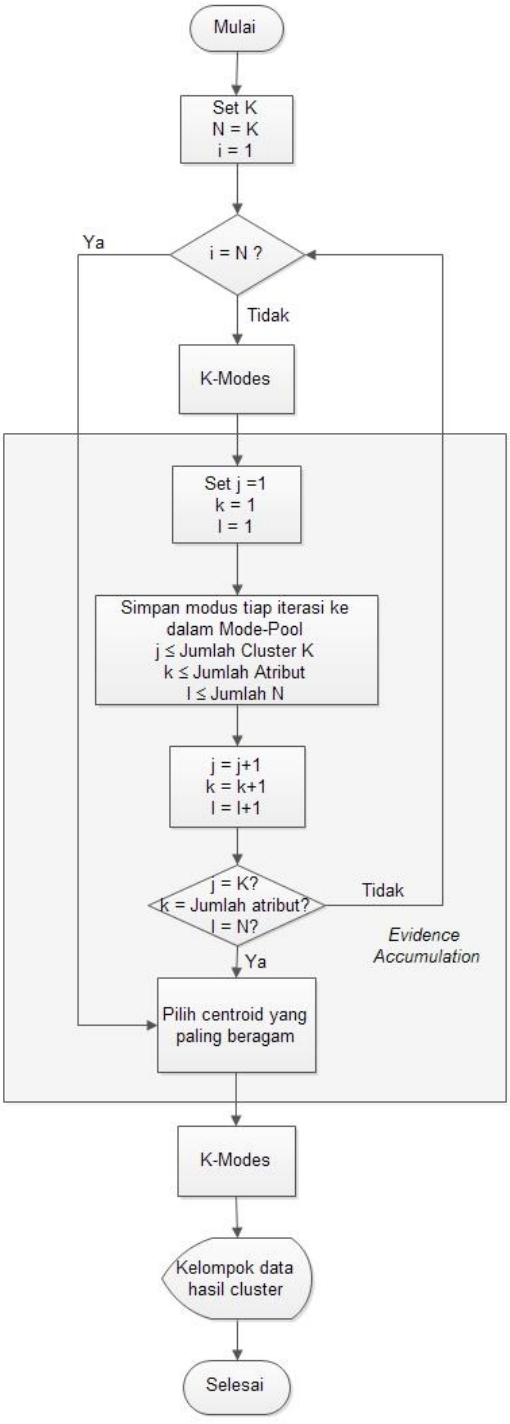
1) Inisialisasi jumlah *cluster* sebanyak k *cluster*.
2) Bangkitkan *initial centroid* (titik pusat) (untuk iterasi pertama secara *random* (acak) dari data yang tersedia dimana banyaknya sesuai dengan *cluster* yang di bentuk (satu titik pusat yang di bentuk untuk tiap *cluster*). sedangkan untuk iterasi 2,3,dst *centroid* diperoleh dari modus (data paling banyak muncul) di setiap atribut per *cluster*nya.
3) Hitung jarak masing-masing data terhadap semua titik pusat *cluster*.
4) Alokasikan setiap objek ke *cluster* dengan modus terdekat menggunakan ukuran jarak *dissimilarity* pada persamaan (2.2).
5) *Update* pusat *cluster* dengan modus (sebagai *centroid*) dari setiap *cluster* dengan nilai kategori yang sering muncul pada setiap *cluster*.
6) Ulangi langkah 2 sampai 5 sampai tidak ada objek yang berpindah *cluster* (*modes* lama berbeda dengan baru) atau perubahan nilai fungsi objektif masih di bawah ambang batas yang ditentukan.



Gambar 2.1 *Flowchart* WDM K-Modes

2.3.2 *Evidence Accumulation*

Algoritma *Evidence Accumulation* merupakan algoritma yang dikembangkan agar dapat menutupi kekurangan pada K-Modes yaitu terhadap penentuan *centroid* awal (Shri dan Shehroz, 2008). Algoritma *Evidence Accumulation* bekerja sebelum dan setelah proses K-Modes. Awalnya, algoritma K-Modes diproses untuk memperoleh modus dari tiap iterasi. Kemudian *Evidence Accumulation* bekerja menyimpan modus di setiap iterasi K-Modes ke dalam *Mode-pool* dan memilih modus yang paling beragam yang dapat dimanfaatkan sebagai *centroid* awal untuk Algoritma K-Modes sehingga algoritma *Evidence Accumulation* mampu menentukan *centroid* awal yang tepat dan menghasilkan akurasi *cluster* yang lebih besar (Shri dan Shehroz, 2008). *Flowchart evidence accumulation* dapat dilihat pada Gambar 2.2



Gambar 2.2 *Flowchart Evidence Accumulation*

2.4 *Purity Measure*

Cluster yang baik akan memiliki persamaan (*similarity*) *intracluster* yang tinggi dan perbedaan (*dissimilarity*) antar *cluster* yang rendah. Setiap *cluster* beranggotakan data yang lebih mirip satu sama lain dalam *cluster* itu dibanding dengan data dari anggota *cluster* yang lain. Untuk mengevaluasi performansi hasil *clustering* yang diperoleh, dapat diukur dengan menggunakan parameter performansi objektif. Untuk mengukur tingkat akurasi *clustering* atau 'r', Huang (1998) mengusulkan cara mengukurnya menggunakan persamaan.

$$r = \frac{1}{n}\sum_{i=1}^{k} a_i \qquad (2.3)$$

Keterangan:

r adalah tingkat akurasi *clustering*

k adalah jumlah *cluster*

a_i adalah objek yang muncul di dalam *cluster* Ci dan pada label *class* yang sesuai.

n adalah banyaknya data yang salah dalam peng*cluster*an ke dalam satu kategori sehingga *n* merupakan total keseluruhan data.

Untuk lebih jelasnya, dapat dilihat contoh pada Tabel 2.1

Tabel 2.1 Hasil pengelompokkan data terhadap kelas sebenarnya

Data	**Kelas Sebenarnya**	**Kelas Hasil Pengelompokan**
D1	A	1
D2	A	1
D3	A	2
D4	B	2
D5	B	1
D6	A	1
D7	B	2

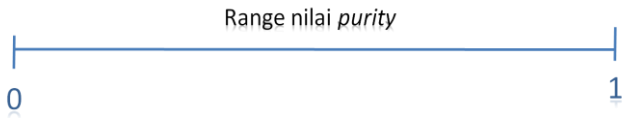
Data berlabel A yang terkelompok pada kelompok 1 sebanyak 3 data, dan yang terkelompok pada kelompok 2 sebanyak 1 data. Sedangkan data B yang terkelompok pada kelompok 1 adalah sebanyak 1 data dan yang terkelompok pada kelompok 2 sebanyak 2 data. Ilustrasinya dapat dilihat di Tabel 2.2

Tabel 2.2 Jumlah data hasil clustering terhadap tabel kelas sebenarnya

Data	Cluster 1	Cluster 2
A	3	1
B	1	2

Cluster 1 dianggap mewakili kelas A dan cluster 2 dianggap mewakili kelas B. Hal ini dikarenakan banyak data pada *cluster* 1 yang terkelompok pada kelas A, sedangkan pada *cluster* 2 lebih banyak data yang terkelompok pada kelas B.

$$\text{Akurasi} = \frac{3+2}{7} = 0.71$$



Gambar 2.3 *Range* nilai *purity*

Berdasarkan Gambar 2.3, jika nilai *purity* semakin mendekati angka 1 maka menunjukkan bahwa *cluster* yang terbentuk merupakan hasil akurasi *clustering* yang semakin baik. Dan sebaliknya, semakin mendekati angka 0 maka hasil akurasi *clustering* semakin buruk.

2.5 Penelitian Yang Relevan

Pada *clustering*, data kategori tidak memungkinkan menggunakan konsep geometri (Z Huang, 1998). Huang mengusulkan algoritma K-Modes untuk mengatasi masalah pada data bersifat kategori yang tidak dapat dihitung

menggunakan konsep geometri. Proses algoritma K-Modes dimulai dengan membentuk terlebih dahulu partisi *cluster* dan menentukan titik pusat di awal proses kemudian secara iteratif. Partisi *cluster* ini terus diperbaiki hingga tidak ada lagi perubahan yang terjadi pada setiap partisi *cluster*. Dilihat dari prosesnya, algoritma K-Modes tergolong algoritma *data mining* yang cepat dan mudah diimplementasikan, Walaupun demikian terdapat beberapa kekurangan yang dimiliki K-Modes salah satunya adalah penentuan titik pusat awal (*initial centroid*) karena penentuan titik pusat awal yang dilakukan secara acak.

Selain sensitif pada penentuan *centroid* awal, algoritma K-Modes juga sensitif terhadap penentuan pengukuran jarak *cluster*. K-Modes menggunakan ukuran *matching* (cocok) atau *mismatching* (tidak cocok) untuk menemukan kesamaan antar objek. *Hamming Distance* atau *Simple Matching Measure* adalah teknik yang digunakan untuk menemukan *similarity* antara dua objek yang akan ditempatkan pada satu *cluster* pada algoritma K-Modes. Namun pada perhitungan ukuran ini tidak mempertimbangkan hubungan kesamaan yang ada pada nilai atribut sehingga menghasilkan kesamaan *intra cluster* yang lemah.

Untuk menemukan kesamaan antar objek pada tipe data kategori, pada penelitian sebelumnya telah memodifikasi algoritma K-Modes untuk memperbaiki dalam pengukuran jarak *cluster* menggunakan *Weighted Dissimilarity Measure* (WDM) sehingga menghasilkan *cluster* lebih akurat dari K-Modes tradisional (Aranganayagi dan Thangavel, 2009). Penelitian sebelumnya telah mengimplementasikan algoritma pada *dataset* dari berjumlah kecil hingga

besar. *Dataset* merupakan himpunan data yang sudah teruji kebenarannya dan sering digunakan pada beberapa penelitian di wilayah *data mining*.

Akan tetapi algoritma WDM K-Modes masih mengadopsi penentuan *centroid* awal yang sama seperti K-Modes, dimana penentuan *centroid* awalnya ditentukan secara acak. Apabila *centroid* awal yang ditentukan sistem tepat maka hasil *cluster* dari WDM K-Modes dapat optimal, sebaliknya jika penentuan *centroid* awal pada WDM K-Modes tidak tepat maka hasil *cluster*nya pun akan kurang optimal. Sehingga dibutuhkan algoritma yang dapat menentukan *centroid* awal yang tepat untuk memperbaiki kinerja algoritma WDM K-Modes.

Salah satu algoritma yang dapat melakukan perbaikan penentuan *centroid* awal agar dapat menghasilkan *centroid* terbaik pada WDM K-Modes adalah *Evidence Accumulation* (EA) (Shri dan Shehroz, 2008). Algoritma *Evidence Accumulation* merupakan algoritma yang dikembangkan agar dapat menutupi kekurangan pada *K-Modes* yaitu terhadap penentuan *centroid* awal sehingga algoritma *Evidence Accumulation* mampu menentukan *centroid* awal yang tepat dan menghasilkan akurasi *cluster* yang lebih besar (Shri dan Shehroz, 2008).

Algoritma *Evidence Accumulation* bekerja sebelum dan setelah proses K-Modes. Awalnya, algoritma K-Modes diproses untuk memperoleh modus dari tiap iterasi. Kemudian *Evidence Accumulation* bekerja menyimpan modus di setiap iterasi K-Modes dan memilih yang paling beragam set modus yang dapat dimanfaatkan sebagai *centroid* awal untuk Algoritma K-Modes.