BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian terkait *data mining* untuk *clustering* maupun *market basket analysis* sudah banyak dilakukan sebelumnya oleh beberapa peneliti dari berbagai instansi, terutama yang berobjek di *minimarket* dan menggunakan metode k-*means* untuk *clustering stock* barang dan metode *fuzzy c-covering* untuk *market basket analysis.* Berikut ini adalah beberapa penelitian yang menggunakan metode k-*means* untuk *clustering* dan *fuzzy c-covering* untuk *market basket analysis* :

1. Penelitian yang dilakukan tahun 2012 oleh Zahedi dan juga Charies Chandra, yang berjudul "Perancangan Program Aplikasi *Market Basket Analysis* Untuk Mendukung Persediaan Barang Dengan Metode *Fuzzy C-Covering*."

   Pada penelitian tersebut, para peneliti mengusung masalah tentang sulitnya menganalisa data transaksi penjualan karena jumlah datanya banyak dan alat pengolah data terbatas yang menyulitkan memperkirakan persediaan barang untuk penjualan berikutnya. Untuk itu dirancangkan aplikasi *market basket analysis* berbasis *fuzzy c-covering*. Nilai *support* dan *confidence* merupakan bagian perhitungan dengan *fuzzy c-covering*. Semakin besar tingkat batasan, *output* dari analisis pun semakin terseleksi [4].
2. Penelitian yang dilakukan pada tahun 2016 oleh Nita Arianty, Oni Soesanto, dan Fatma Indriani, berjudul "Penerapan Metode *Fuzzy c-Covering* Untuk Analisis Pola Pembelian Pada *Minimarket*."

   Pada penelitian tersebut, peneliti mengusung masalah tentang proses *data mining* pada analisa pola pembelian di *minimarket* berbasis *fuzzy c-covering* untuk mencari nilai *support* dan nilai *confidence*. Metode *fuzzy c-covering* ini akan mengklasifikasi elemen-elemen himpunan universal yakni keseluruhan *item* pada *minimarket* menjadi partisi-partisi yang lebih fokus dan detail berdasarkan jenis produk *item* apa saja yang ada. Karena itu *fuzzy c-covering*

diterapkan agar dapat mengatasi hambatan yang selama ini terjadi pada proses *market basket* di dalam mencari hubungan antar *item* [5].

3. Penelitian yang dilakukan pada tahun 2015 oleh Benri Melpa Metisen dan Herlina Latipa Sari, berjudul "Analisis *Clustering* Menggunakan Metode K-*Means* Dalam Pengelompokkan Penjualan Produk Pada Swalayan Fadhila."

   Pada penelitian tersebut, para peneliti mengusung masalah tentang swalayan Fadhila yang bisnisnya masih dengan cara manual sehingga sering terjadi kesalahan pencatatan data. Selain itu juga tidak dapat mengelompokkan produk antara yang laris atau kurang laku, sehingga sulit menentukan *stock*. Untuk itu dibutuhkan sistem *clustering* yang mana menggunakan algoritma k-*means* agar dapat menunjang kebutuhan bisnis dari proses tersebut [6].
4. Penelitian yang masuk di dalam sebuah proceedings Seminar Nasional Riset Ilmu Komputer Universitas Muslim Indonesia Makassar tahun 2016, dengan peneliti Warnia Nengsih berjudul "*Descriptive Modelling* Menggunakan K-*Means* untuk Peng*cluster*an Tingkat Kemiskinan di Propinsi Riau."

   Penelitian tersebut membahas tentang kemiskinan Propinsi Riau yang terus meningkat sehingga perlu ditangani oleh pemerintah. Namun karena adanya persebaran daerah tingkat keparahan kemiskinan sangat beragam, sehingga terjadi kesulitan menentukan wilayah mana yang kemiskinan paling tinggi. Untuk itu diperlukan metode k-*means* yang mampu meng-*cluster* wilayah-wilayah tersebut masuk ke dalam tingkat kemiskinan yang mana [7].
5. Penelitian yang dilakukan pada tahun 2015 oleh Elly Muningsih dan Sri Kiswati, berjudul "Penerapan Metode K-*Means* untuk *Clustering* Produk *Online Shop* dalam Penentuan Stok Barang."

   Penelitian tersebut, para peneliti mengusung masalah tentang bagaimana permintaan konsumen yang fluktuatif sehingga persediaan *stock* yang tidak stabil. Pada *online shop* tersebut ada beberapa kesulitan yang dialami seperti menentukan *stock* minimun tiap barang, yang harus dipenuhi sesuai minat konsumen. Untuk menangani masalah tersebut, diperlukan sebuah metode yang dapat mengelompokkan tingkat *stock* barang yang banyak, sedang, atau sedikit menggunakan algoritma k-*means clustering* [8].

Tabel 2.1 Penelitian Terkait

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1.	- Zahedi - Charies Chandra (Tahun 2012)	Menganalisa data transaksi penjualan yang banyak dan alat pengolah data terbatas sehingga menyulitkan memperkirakan persediaan barang dijual berikutnya.	*Fuzzy C-Covering*	Program aplikasi yang digunakan perusahaan melakukan analisa data guna memperkirakan persediaan barang untuk masa mendatang [4].
2.	- Nita Arianty - Oni Soesanto - Fatma Indriani (Tahun 2016)	Klasifikasi elemen dari himpunan keseluruhan *item* menjadi partisi yang lebih fokus, detail berdasar jenis *item* agar mengatasi proses *market basket* di dalam mencari asosiasi antar *item*.	*Fuzzy C-Covering*	Dari data sampel, nilai *confidence* paling tinggi adalah "jika peralatan bayi maka susu" (*support* = 1,13181% dan *confidence* = 38,70967%) [5].
3.	- Benri Melpa Metisen - Herlina Latipa Sari (Tahun 2015)	Analisa data *clustering* produk untuk mengatasi kesulitan penentuan tingkat *stock* barang, sekaligus merancang sistem aplikasi pendukung *clustering*-nya.	*K-Means*	Dari data sampel didapat 2 jenis kelompok data, yaitu dengan penjualan rendah dan tinggi. Sehingga diketahu mana produk laris / tidak [6].
4.	- Warnia Nengsih (Tahun 2016)	Analisa wilayah persebaran tingkat kemiskinan yang terjadi di Propinsi Riau yang sangat beragam, perlu dilakukan *clustering* untuk mengetahui tingkatannya.	*K-Means*	Dari 12 kecamatan yang di-*cluster*, didapatkan 3 kelompok *cluster* dimana *cluster* 2 terdapat ada 2 *record*, *cluster* 3 ada 10 *record*, sedang *cluster* 1 tidak ada *record* [7].

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
5.	- Elly Muningsih - Sri Kiswati (Tahun 2015)	Terjadinya kesulitan *online shop* dalam menentukan *stock* minimum tiap barang sesuai minat konsumen, sehingga sering terjadinya kesalahan perkiraan atau *stock* yang tidak akurat.	*K-Means*	Rancangan aplikasi yang dapat mengelompokkan produk menjadi 3 *cluter* dengan kategori *stock* banyak, sedang, dan sedikit berdasarkan pada transaksi penjualan [8].

Dari beberapa penelitian yang sudah dicantumkan di atas, dapat diketahui bahwa penggunaan algoritma k-*means* dapat memudahkan dalam melakukan *clustering* terhadap ribuan bahkan jutaan data sehingga diperoleh informasi yang mulanya tidak nampak menjadi nampak yang dapat digunakan untuk menindaklanjuti suatu penyelesaian masalah. Sedangkan untuk penggunaan algoritma *fuzzy c-covering* dapat memudahkan dalam mencari *association rule* antar *item*, mana *item* paling sering keluar secara bersamaan di suatu transaksi dari banyaknya transaksi yang pernah terjadi atau sering dikenal dengan analisa keranjang belanja.

2.2 Data Mining

2.2.1 Konsep *Data Mining*

Ada banyak buku membahas tentang *data mining*, dan setiap penulis mempunyai definisinya masing-masing. Seperti buku yang ditulis oleh Gordon S. Linoff dan Michael J. A. Berry berjudul "*Data Mining Techniques For Marketing, Sales, and Customer Relationship Management*", mendefinisikan tentang *data mining* adalah sebuah proses bisnis untuk mengeksplorasi data-data yang berjumlah besar untuk menemukan pola dan kaidah berarti [9]. Definisi lain tentang *data mining* dalam buku yang ditulis oleh Charu C. Aggarwal berjudul "*Data Mining The Textbook*", mendefinisikan bahwa *data mining* merupakan pembelajaran dari mengoleksi, membersihkan, menganalisa, dan memperoleh beberapa pengetahuan yang sangat

berguna dari data [10]. Sedangkan definisi menurut Ian H. Witten dan kawan-kawan dalam bukunya yang berjudul "*Data Mining Practical Machine Learning Tools and Techniques*", mendefinisikan *data mining* adalah kegiatan menemukan pola-pola dari banyak data. Proses harus otomatis atau semiotomatis. Pola yang ditemukan harus berarti karena menghasilkan beberapa keuntungan [11].

Dilihat dari beberapa definisi tentang *data mining* yang telah dijelaskan oleh para penulis dalam buku-buku mereka, sebenarnya mempunyai maksud sama. Bahwa *data mining* merupakan proses mengeksplorasi data berjumlah besar untuk dicari pola-polanya sehingga menjadi pengetahuan yang mempunyai arti dan berguna.

Secara garis besarnya tentang metode, *data mining* dibedakan dalam 2 pendekatan diantaranya *Supervised Learning* yaitu metode yang mencoba mencari hubungan antara atribut yang bervariable bebas dan sebuah atribut bervariabel tidak bebas. Hubungan dapat ditemukan dan diwakili dalam struktur yang disebut model. Biasanya model menjelaskan fenomena tersembunyi dalam *dataset* dan digunakan untuk memprediksi nilai atribut target hanya dengan mengetahui nilai-nilai atribut *input* [12]. Sedangkan *Unsupervised Learning* yaitu kebalikan dari *supervised learning,* di mana label masing-masing data pelatihan tidak diketahui sebelumnya.

Berdasarkan pada tugas yang dapat dilakukan oleh *data mining* untuk mencari pola-pola data, dibedakan menjadi beberapa kelompok yaitu [13] :

1. *Description,* merupakan penggambaran pola dan kecenderungan yang ada di dalam data secara sederhana. Deskripsi tersebut sering memberi kemungkinan penggambaran untuk sebuah pola atau kecenderungan.
2. *Classification,* teknik memperhatikan tingkah laku dan atribut himpunan yang sudah didefinisikan. Teknik ini memberi klasifikasi untuk data baru dengan manipulasi data yang telah terklasifikasi dan *output*-nya berguna memberikan sejumlah aturan. Dalam kasusnya, klasifikasi merupakan *supervised learning*.
3. *Prediction,* hampir sama seperti klasifikasi dan estimasi. Namun nilai *output* prediksi akan adanya di masa mendatang. Beberapa teknik untuk klasifikasi juga estimasi bisa digunakan (dalam keadaan tepat) di dalam prediksi. Dalam kasusnya, prediksi merupakan *supervised learning*.

4. *Estimation,* hampir mirip klasifikasi. Namun variabel targetnya lebih bersifat numerik daripada kategori. Model dibangun berdasarkan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Dalam kasusnya, estimasi merupakan *supervised learning*.
5. *Clustering,* adalah pengelompokan *record* atau pengamatan juga menciptakan kelas objek-objek yang ada kemiripan antara satu dengan lain dan memiliki ketidakmiripan dengan *record* dari *cluster* lain. Pada kasusnya, *clustering* merupakan *unsupervised learning*.
6. *Association Rule,* tugas asosiasi yang disebut juga "*market basket analysis*" adalah mencari relasi antara himpunan *item* dan menemukan atribut yang sering muncul dalam satu waktu dari banyak waktu yang pernah terjadi. Dalam kasusnya, asosiasi menggunakan metode *unsupervised learning*.

2.2.2 Konsep *Clustering*

Clustering merupakan proses melakukan partisi atau pemisahan satu set *item* data ke himpunan bagian yang disebut *cluster*. *Item* yang dalam *cluster*-nya memiliki karakteristik hampir sama antara satu dengan yang lain dan akan berbeda dengan *item* dalam *cluster* lain. Partisi tidak dilakukan secara manual karena pasti akan sulit mengingat banyaknya data yang akan dipartisi, sehingga dibutuhkan suatu algoritma *clustering*. Oleh karena itu, *clustering* sangat berguna terlebih dalam menemukan kelompok yang tidak dikenal dalam data. Di *bussniness inteligence* khususnya *minimarket*, *clustering* membantu mengatur banyaknya *item* barang yang dijual menjadi beberapa kelompok. Contoh mengelompokkan *item* barang dalam beberapa *cluster* dengan kesamaan karakteristik seperti *item* barang mana yang kurang laku, laku, atau sangat laku. *Clustering* juga dikenal sebagai data segmentasi karena mempartisi banyak data ke banyak grup berdasar kesamaan. Berikut beberapa manfaat yang diberikan oleh proses *clustering* [14] :

1. *Clustering* adalah metode segmentasi data yang berguna dalam memprediksi dan menganalisa masalah bisnis tertentu, misalnya segmentasi pasar
2. Identifikasi objek dalam berbagai bidang penelitian seperti *computer vision* dan *image processing*.

Sebetulnya ada beberapa tipe *clustering* yang dapat digunakan dalam melakukan pengelompokkan data sesuai kebutuhan dan tujuan, berikut tipe-tipe *clustering* :

1. *Partitioning algorithms*, merupakan algoritma yang cara kerjanya membentuk berbagai macam partisi lalu mengevaluasinya berdasarkan kriteria tertentu.
2. *Hierarchial algorithms*, merupakan algoritma yang membentuk dekomposisi hirarki dari sekumpulan data menggunakan beberapa kriteria.
3. *Density based,* merupakan algoritma yang membentuk *cluster* berdasar pada koneksi dan fungsi densitas.
4. *Grid-based,* membentuk *cluster* berdasar struktur *multiple level granularity*.
5. *Model-based,* menganggap jika sebuah model adalah hipotesa untuk masing-masing *cluster* dan model yang baik dipilih diantara model hipotesa tersebut.

Dalam melakukan proses *clustering* pun ada syarat-syaratnya sekaligus tantangan yang harus dipenuhi oleh suatu algoritma *clustering*, menurut Jiawei Han dan Micheline Kamber di dalam bukunya yang berjudul "*Data Mining Concepts And Techniques*" yang dijelaskan sebagai berikut [15] :

1. Skalabilitas, yang artinya mengharuskan bahwa suatu metode *clustering* harus mampu menangani data dalam jumlah yang besar bahkan sampai jutaan data.
2. Kemampuan analisa beragam bentuk data, yang artinya algoritma *clustering* harus mampu diimplementasikan ke berbagai macam bentuk data seperti data nominal, ordinal maupun gabungannya.
3. Menemukan *cluster* dengan bentuk-bentuk yang tidak terduga, artinya hasil dari *clustering* dapat berbentuk aneh dan tidak sama antara satu data dengan lainnya. Karena itu butuh kemampuan analisa *cluster* dengan bentuk apapun pada suatu algoritma *clustering*.
4. Kemampuan untuk dapat menangani *noise*, yang artinya data itu tidak selalu dalam keadaan baik. Terkadang ada data rusak, tidak dimengerti, atau hilang. Karena itu, algoritma *clustering* harus mampu menangani data yang rusak.
5. Sensitifitas pada perubahan *input*, artinya perubahan *input* data menyebabkan perubahan *cluster* yang ada bahkan menyebabkan perubahan mencolok jika menggunakan algoritma *clustering* yang memiliki tingkat sensitifitas rendah.

6. Mampu meng-*clustering* data-data berdimensi tinggi, artinya suatu kelompok data dapat berisi banyak dimensi atau atribut. Untuk itu algoritma *clustering* harus mampu menangani data berdimensi tinggi tersebut.
7. Interpresasi dan kegunaan, artinya mengharuskan bahwa hasil dari proses *clustering* dapat diinterpretasikan dan juga berguna.

2.2.3 Konsep *Market Basket Analysis*

Association rule merupakan bentuk dari suatu formula jika "kejadian sebelumnya" kemudian "konsekuensinya" (*IF antecedent, THEN consequent*). *Association rule* terkenal karena aplikasi analisa keranjang belanja, sehingga sering disebut *market basket analysis*. Tugas *association rule* adalah mencari pola dari banyak data yang ada yang mempunyai pola asosiasi antara dua atau lebih *item*. Dalam *association rule*, terdapat ukuran keterkaitan yang didapat dari mengolah data menggunakan proses hitung tertentu. Secara umum ada 2 ukuran tersebut, diantaranya :

1. *Support*, ukuran yang mempresentasikan besaran tingkat dominasi *item* atau *item*set dari seluruh transaksi untuk memutuskan apakah *item* itu layak dicari *confidence*-nya (misalnya dari kesemua total transaksi, berapa besaran tingkat dominasi yang mempresentasikan *item* X dan Z dibeli secara bersama).
2. *Confidence*, ukuran yang menunjukkan hubungan 2 *item* secara *conditional* (misalnya, seberapa sering *item* X dibeli jika orang membeli *item* Z).

Kedua ukuran tersebut nanti bermanfaat saat mencari *interesting association rule*, yaitu guna dibandingkan dengan pembatas (*threshold*) yang ditentukan *user*. Batas tersebut biasanya terdiri dari *min_support* dan *min_confidence*.

Market basket analysis mengacu pada berbagai metode yang mempelajari tentang *item-item* di suatu keranjang belanja yang dibeli pada satu penotalan dari banyak kejadian transaksi yang telah terjadi. Data keranjang belanja dalam bentuk yang paling mentah merupakan daftar transaksi belanja yang dilakukan oleh pelanggan, yang mengidentifikasikan hanya barang-barang yang dibeli secara bersamaan oleh ribuan pelanggan. Datanya ini menantang karena beberapa hal [16] :

1. Jumlah data yang digunakan dalam jumlah besar (biasanya dapat mencapai jutaan transaksi per hari jika pangsa pasarnya besar, seperti pada swalayan).
2. Pembagiannya barangnya luas (setiap keranjang belanja hanya mengandung sebagian kecil *item* dari ribuan *item* yang dijual).
3. Heterogenitas (orang-orang dengan selera yang berbeda cenderung membeli sekelompok barang yang spesifik).

Tujuan analisis keranjang belanja adalah mengidentifikasi produk-produk mana yang cenderung dibeli secara bersamaan. Dengan menganalisis data pada tingkat transaksi maka akan dapat diidentifikasikan pola-pola pembelian, seperti misalnya jenis camilan ringan mana dan jenis roti mana yang dibeli secara bersama dengan minuman untuk bersantai jika orang ingin mengadakan suatu acara seperti piknik atau yang sering dilakukan ketika *event-event* sepakbola yang biasanya dilakukan nonton bareng. Informasi tersebut kemudian dapat digunakan untuk menentukan di mana suatu *item* dengan tingkat asosiasi tertentu dengan *item* lain ditempatkan pada rak-rak di toko yang memungkinkan *item-item* tersebut saling berdekatan, dan hal ini juga akan dapat membantu manajemen persediaan.

2.3 Algoritma K-*Means*

2.3.1 Konsep K-*Means*

Algoritma K-*Means* merupakan salah satu algoritma *clustering* pada *data mining* non hierarki yang menganalisa data dengan pemodelan *unsupervised learning* dan melakukan pengelompokan data dengan sistem partisi. Konsep dalam *clustering* yaitu dengan mengelompokkan data-data menjadi beberapa himpunan, yang mana data dalam satu kelompok mempunyai karakteristik yang sama antara data satu dengan yang lainnya dan mempunyai karakteristik berbeda dengan data yang ada dalam kelompok berbeda. Dengan maksud lain, metode k-*means* ini melakukan peminimalan variasi antar data yang ada dalam suatu *cluster* dan memaksimalkan variasi dengan data yang ada dalam *cluster* lainnya. Dalam mengelompokkan data dengan k-*means*, didasarkan pada titik pusat yang telah ditentukan sebelumnya. Dengan menggunakan rumus *Euclidean Distances, Clustering* objek dilihat dari

jarak objek dengan titik pusatnya yang paling dekat, yang mana dalam mencari jarak terdekat kadang dibutuhkan hingga beberapa perulangan atau iterasi sampai objek benar-benar mendekati ke titik pusat *cluster*. *Euclidean Distances* sendiri merupakan perhitungan jarak dari 2 titik di *euclidean space,* dengan menghitung akar dari kuadrat perbedaan 2 vektor. Setelah diketahui titik pusat terdekatnya dan posisi objek terhadap *cluster*-nya tidak berubah-ubah lagi, objek dapat dipastikan masuk sebagai anggota *cluster* tersebut. Sehingga diketahui bagaimana k-*means* mempartisi data-data ke dalam kelompok masing-masing *cluster*, melalui proses perubahan posisi data yang awalnya bergejolak menjadi tetap. Beberapa kelebihan dari algoritma k-*means* antara lain sebagai berikut :

1. Karena algoritma k-*means* lebih sederhana dibandingkan algoritma lain seperti *fuzzy c-means*, maka dalam implementasinya mudah dilakukan.
2. Dalam mempelajari k-*means* relatif cepat sehingga tidak perlu waktu lama.
3. Mudah untuk diadaptasikan dari satu masalah ke masalah lain selama data-data yang digunakan juga cocok dilakukan peng-*cluster*-an.
4. Algoritma k-*means* sudah begitu banyak digunakan dalam berbagai masalah, terlebih dalam penelitian-penelitian terdahulu.

Selain algoritma k-*means* itu sendiri, ternyata banyak algoritma yang didapat dari hasil pengembangan atau variasi dari metode k-*means* sebagai berikut [17] :

1. K-*means*++, merupakan metode untuk menentukan nilai awal pada k-*means*. Algoritma ini diperlukan dalam meminimalkan dampak buruk dari metode k-*means* yang begitu bergantungan dari nilai awalnya.
2. K-*medoids*, merupakan algoritma yang berbasis *prototype*. Jika k-*means* perlu titik tengah sebagai model dari peng-*cluster*-annya, maka untuk k-*medoids* ini menggunakan rata-rata tengah.
3. *Bisecting* k-*means*, dasarnya memerlukan k-*means* sebagai pembagi 2 *cluster*. Mulanya tiap-tiap objek tergabung dalam suatu *cluster*. Di tiap iterasi, dipilih satu *cluster* untuk dibagikan 2 menggunakan k-*means*. Ini dilakukan sampai tercipta k-*cluster*. Algoritma *bisecting* k-*means* kerjanya lebih cepat dibanding k-*means* karena mereduksi jumlah objek yang dibandingkan di tiap iterasinya.

2.3.2 Proses Algoritma K-*Means*

Untuk menemukan pola-pola data dari proses *clustering* berbasis k-*means*, berikut langkah-langkah yang perlu dilakukan [6], [7], [8] :

1. Menentukan dahulu berapa jumlah *cluster* yang akan dibentuk (k-*cluster*).
2. Menentukan titik pusat *cluster (centroid)* secara acak.

   Untuk menghasilkan *cluster* maksimal, titik awal partisi merupakan salah satu faktor yang berpengaruh untuk itu pemilihan titik awal harus beralasan. Dalam menentukan titik pusat awal *cluster* dapat dengan mengukur nilai titik tengah segmentasi berdasarkan jumlah jarak terpendek antar anggota kelas tersebut.
3. Menghitung jarak antara setiap data ke titik pusat terdekat.

   Distance space digunakan dalam perhitungan jarak antara data dan titik pusat. Salah satu persamaan yang dapat digunakan adalah *Euclidean Distance Space*, yang sering digunakan dalam menghitung jarak. Hal tersebut dikarenakan hasil perhitungan merupakan jarak terpendek antara dua titik yang diperhitungkan. Berikut ini merupakan persamaannya yaitu :

   $$Dn\ (x,y) = \sqrt{\sum_{k=1}^{p}\left(X_{xk} - Y_{yk}\right)^2} \qquad (2.1)$$

   Dengan :

   Dn (x,y) = jarak objek antara objek i dan j

   P = dimensi data

   X_{xk} = koordinat dari objek x pada dimensi k

   Y_{yk} = koordinat dari objek y pada dimensi k
4. Mengalokasikan masing-masing data ke titik pusat terdekat, sehingga dalam tahap ini setiap data sudah mulai bergerak ke *cluster*-nya masing-masing.
5. Menghitung titik pusat baru menggunakan jumlah keseluruhan per *cluster.*

   Untuk menghitung titik pusat *cluster* yang baru ditentukan berdasarkan rata-rata dari data-data yang ada di titik pusat yang sama. Berikut persamaannya :

   $$C_k = \left(\frac{1}{n_k}\right)\sum d_i \qquad (2.2)$$

Dengan :

C_k = titik pusat baru

n_k = jumlah dokumen dalam *cluster* k

d_i = dokumen dalam *cluster* k

6. Kembali ke langkah 3, jika ada data yang berubah-ubah dari *cluster* atau jika berubahnya nilai *centroid* melebihi nilai *threshold* atau jika berubahnya nilai *objective function* yang digunakan melebihi nilai *threshold* yang digunakan.

2.4 Algoritma *Fuzzy C-Covering*

2.4.1 Konsep *Fuzzy C-Covering*

Fuzzy c-covering merupakan suatu algoritma yang digunakan dalam klasifikasi elemen-elemen dari himpunan universal ke dalam partisi-partisi berupa *fuzzy sets*. *Fuzzy c-covering* merupakan bentuk generalisasi dari metode *fuzzy* c-*partition* yang mana *fuzzy* c-*partition* diidentifikasikan sebagai berikut [4] :

Misal I= $\{i_1, i_2, \dots, i_n\}$ adalah domain data. *Fuzzy* c-*partition* dari I adalah *fuzzy subset* atau *fuzzy classes* dari T, ditandai oleh T= $\{t_1, t_2, \dots, t_c\}$ yang memenuhi :

$$\sum_{m=1}^{c} \mu\, t_m\,(i_k) = 1, \text{ untuk semua } k \in N_n \qquad (2.3)$$

Sedangkan berikutnya

$$0 < \sum_{k=1}^{n} \mu\, t_m\,(i_k) < n, \text{ untuk semua } m \in N_c \qquad (2.4)$$

Di mana c merupakan *positive integer* ($0 < c \leq n$) dan $\mu t_m(i_k) \in [0,1]$.

Untuk lebih jelasnya rumus (2.3) dan (2.4), berikut contoh 1 :

$\mu_{t1} = \{0.3/i_1,\ 0.7/i_2,\ 1/i_3\}$,

$\mu_{t2} = \{0.6/i_2,\ 0.5/i_3,\ 1/i_4\}$,

$\mu_{t3} = \{0.3/i_4,\ 0.6/i_5,\ 0.7/i_6\}$.

Di contoh 1 diketahui total elemen terlibat (n = 6) dan partisinya (c = 3). *Fuzzy class* tersebut bisa diistilahkan sebagai *fuzzy* c-*partition* dari suatu domain I = $\{i_1, i_2, i_3, i_4, i_5, i_6\}$ karena memenuhi rumus (2.3) dan (2.4). Di dalam *fuzzy c-covering*, rumus (2.3) dari teori *fuzzy* c-*partition* digeneralisasikan menjadi :

$$\sum_{m=1}^{c} \mu\, t_m\,(i_k) \geq 1, \text{ untuk semua k} \in N_n \qquad (2.5)$$

Untuk memperjelasnya diberikan contoh ke 2, yang misalnya suatu *fuzzy classes* disebutkan seperti di bawah ini :

$\mu_{t1} = \{0.3/i_1,\ 0.7/i_2,\ 0.6/i_3\}$,

$\mu_{t2} = \{0.7/i_6,\ 0.5/i_3,\ 1/i_4,\ 0.3/i_5\}$,

$\mu_{t3} = \{0.3/i_4,\ 0.6/i_1,\ 0.7/i_5\}$.

Dari contoh 2 bisa diketahui (n = 6) dan jumlah partisinya (c = 3). *Fuzzy class* itu dikatakan sebagai *fuzzy c-covering* suatu domain I = $\{i_1, i_2, i_3, i_4, i_5, i_6\}$ karena telah memenuhi rumus (2.4) serta (2.5).

2.4.2 Proses Algoritma *Fuzzy C-Covering*

Untuk menemukan pola data dari proses *market basket analysis* berbasis *fuzzy c-covering*, berikut langkah-langkah yang perlu dilakukan [4], [5]:

1. Menentukan *max_item_threshold* sesuai yang dibutuhkan. *Max_item_threshold* merupakan pembatas untuk menyaring transaksi berdasar pada jumlah *item* di dalam transaksi. Hal ini didasarikan pada pengetahuan bahwa makin banyak *item* yang dibeli dalam suatu transaksi, hubungan antar *item* dalam transaksi tersebut pun akan semakin lemah.
2. Mencari *record-record* yang berada di dalam tabel transaksi yang memenuhi *max_item_threshold* dan menyimpannya ke dalam QT, di mana :

   $$QT = \{t \mid |t| \leq ith,\ ith \in \textit{positive integer}\} \qquad (2.6)$$

   Dengan :

   QT (*Qualified Transaction*) = himpunan transaksi yang sudah memenuhi *max_item_threshold*

   t = transaksi

   |t| = jumlah produk dalam suatu transaksi

   ith = *max_item_threshold*

3. Set k = 1 (k adalah variabel untuk menentukan jumlah kombinasi).

4. Menentukan *min_support* ke-k yang sebagai *threshold* bagi kombinasi k-*item* terhadap tingkat dominasinya dari keseluruhan transaksi.
5. Mencari *support* dari setiap kombinasi k-*item* yang memungkinkan yang ada di dalam transaksi tersebut dengan rumus :

$$support\ (u) = \frac{\sum_{t=1}^{n} \frac{1}{C_{|T_t|}^{k}\ s(u,\ T_t)}}{n} = \frac{\sum_{t=1}^{n} \frac{1}{\frac{|T_t|!}{k!(|T_t|-k)!}\ s(u,\ T_t)}}{n} \qquad (2.7)$$

Dengan :

u = kombinasi k-*item* yang akan dicari *support*-nya. Jika I merupakan universal *set* dari *item-item*, u $\subseteq I$

|u| = k : jumlah produk dalam u

T_t = transaksi ke-t ($T_t \subseteq I$)

$|T_t|$ = jumlah produk dalam T_t

$C_{|T_t|}^{k}$ = kombinasi k-*item* terhadap $|T_t|$

n = jumlah *record* atau tuple dalam QT

$s(u, T_t) \in \{0, 1\}$ adalah suatu *function*, di mana jika u $\in T_t$, $s(u, T_t) = 1$, selain itu $s(u, T_t) = 0$.

6. Melakukan penyaringan terhadap kombinasi produk yang ada dalam transaksi tersebut yang tidak memenuhi : *support* (u) ≥ *min_support* ke-k.
7. *Set* k = k + 1, di mana jika k > ith, maka ke langkah 9 (sembilan).
8. Mencari kombinasi k-*item* yang memungkinkan dari tiap kombinasi (k-1)-*item* yang memenuhi *minimum_support* yang telah ditentukan, dengan cara :
   a. Untuk mendapat kombinasi k-*item*, u, harus ada kombinasi (k-1)-*item*, u', di mana u' $\subset$ u. Misalnya untuk mendapatkan u = $\{I_1, I_2, I_3, I_4\}$ harus ada u' = $\{I_1, I_2, I_3\}$, $\{I_1, I_2, I_4\}$, $\{I_1, I_3, I_4\}$, $\{I_2, I_3, I_4\}$.
   b. Jika tidak ada lagi kombinasi k-*item* yang munkin memenuhi *min_support* yang ditentukan, dapat ke langkah 9, selain itu ulangi langkah 4 sampai 7.
9. Mendefinisikan tiap produk yang telah didapatkan dari langkah sebelumnya sebagai *fuzzy set* atau juga disebut *item fuzzy set* terhadap transaksi QT.

10. Mencari *candidate rules* dengan menghitung *confidence* di tiap kombinasi k-*item* yang mencapai *min_support* ke-k (k ≥ 2) dari *item fuzzy set* yang sudah didapatkan dari langkah 9 (sembilan) dengan rumus :

$$R(X, Y) = \textit{confidence}\ (Y \rightarrow X) = \frac{\sum_{t \in T}\ i \in \inf_{X} \cup\, Y\ (\mu_i(t))}{\sum_{t \in T}\ i \inf_{\in}\ Y\ (\mu_i(t))} \qquad (2.8)$$

Dengan :

$X, Y \subseteq I$

T = himpunan kode-kode transaksi yang terdapat di QT

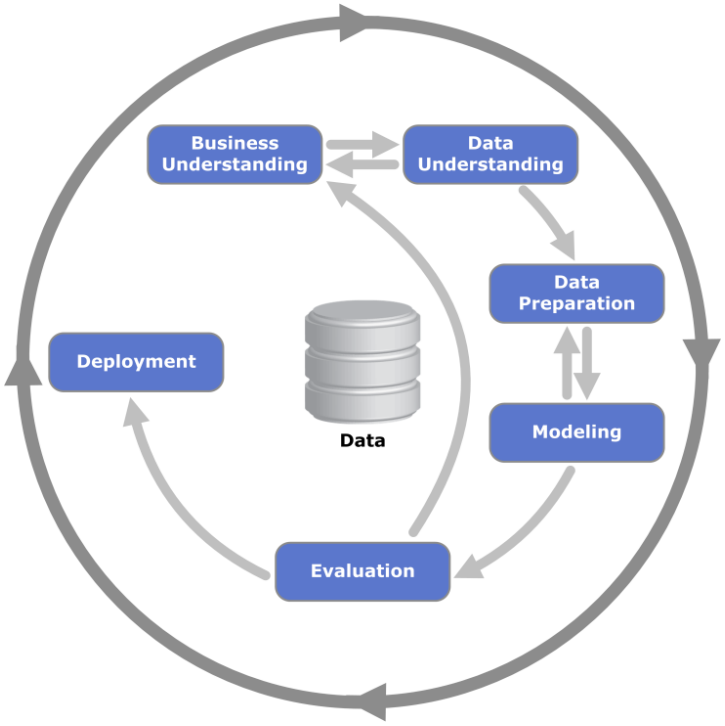
$\mu_i(t) \in [0, 1]$ = fungsi anggota terhadap T

2.5 CRISP-DM (*Cross Industry Standard Process For Data Mining*)

2.5.1 Tentang CRISP-DM

CRISP-DM (*Cross Industry Standard Process For Data Mining*) adalah sebuah metodologi yang menyediakan sebuah pendekatan terstruktur untuk perencanaan proyek *data mining*, yang merupakan suatu konsorsium perusahaan yang didirikan oleh Komisi Eropa tahun 1996 dan telah ditetapkan sebagai proses standar dalam *data mining* yang dapat diaplikasikan di berbagai sektor industri. Adapun proses ini salah satu tujuannya ialah menemukan pola yang menarik dan bermakna dalam data, serta melibatkan beberapa disiplin ilmu seperti statistika, *machine learning*, *artificial intelligence*, dan *data mining*. Salah satu keuntungan dari menggunakan proses ini adalah menjelaskan langkah paling umum dalam proses-prosesnya yang bukanlah suatu prosedur yang kaku dan harus dilakukan secara beruntun. Bahkan beberapa kasus ada yang dilakukan tidak dari awal tahap, bisa jadi dari pemodelan dahulu atau dari mengenali data-data yang dimiliki terlebih dahulu. CRISP-DM juga tidak hanya bisa dilakukan oleh para analis ahli, karena prosesnya yang tidak terlalu sulit maka para pemula atau pelajar pun dapat menggunakannya. Namun meski demikian, CRISP-DM memberikan kerangka kerja yang berguna dalam penggalian data sehingga lebih tahu tentang apa saja yang perlu dilakukan. Untuk lebih jelasnya bagaimana fase-fase CRISP-DM dan apa saja yang biasa dilakukan juga diperlukan dalam setiap fase, akan dijelaskan pada poin selanjutnya.

2.5.2 Proses CRISP-DM



Gambar 2.1 Proses Diagram CRISP-DM
(*Cross Industry Standard Process For Data Mining*)[18]

Dilihat dari gambar, nampak terdapat beberapa proses atau fase-fase yang perlu dilakukan pada analisa *data mining* menggunakan CRISP-DM. Secara garis besar ada 6 alur proses yang mana proses pertama dimulai dari *business understanding*, lalu *data understanding* yang perlu dianalisa berdasarkan *business understanding* yang jika sudah sesuai maka belanjut ke *data preparation*, kemudian *modelling*. Pada tahap *modelling* sangat erat kaitannya dengan *data preparation*, sehingga di tahap *data preparation* tidak boleh ada kesalahan saat menyiapkan data-datanya. Setelah *modeling* selesai, perlu dilakukan evaluasi terhadap hasil dari modeling. Dari evaluasi dapat berlanjut ke 2 arah, yaitu siap untuk *deployment* atau diulang kembali ke tahap *business understanding*. Untuk pengulangan bisa dikarenakan perlu adanya penelitian ulang terhadap keseluruhan proses yang sudah dilakukan, atau melakukan kembali analisa *data mining* dengan *business understanding* yang berbeda, atau juga sebagai siklus hidup sebuah sistem informasi. Berikut ini akan dijelaskan lebih mendalam setiap tahap proses CRISP-DM [19] :

1. *Business Understanding*

   Dinamakan juga sebagai pemahaman penelitian. Tahap ini sangat diperlukan kapasitas pemahaman yang bagus tentang substansi dari kegiatan *data mining* yang akan dilakukan, atau merupakan kebutuhan dari sudut pandang bisnis. *Business understanding* terdiri dari beberapa sub tahapan, yang diantaranya yaitu: menentukan tujuan atau sasaran bisnis, memahami proses bisnis yang sedang berjalan, menentukan tujuan dari *datamining*, dan pengaturan rencana strategi serta jadwal pelaksanaan kegiatan.

2. *Data Understanding*

   Merupakan fase di mana data-data mentah atau data awal dikumpulkan, lalu memahami data-data tersebut untuk dapat mengenal data apa saja yang akan digunakan dalam proses *data mining*. Fase ini mencoba mengidentifikasikan masalah apa saja yang berkaitan dengan kualitas data, mendeteksi *subset* yang menarik dari data untuk membuat hipotesa awal.

3. *Data Preparation*

   Disebut juga sebagai persiapan data, yang mana tahapan ini lah yang dapat dikatakan paling rumit atau banyak membutuhkan waktu karena ada banyak sub-sub fase yang memang perlu dilakukan seperti *cleaning* data, *integrating*, *selecting*, bahkan pembuatan atribut baru jika dibutuhkan, dan sebagainya.

4. *Modeling*

   Merupakan fase penentuan tehnik atau algoritma *data mining* yang akan digunakan, *tools* yang dapat digunakan untuk mendukung algoritma, dan juga parameter dengan nilai yang optimal.

5. *Evaluation*

   Merupakan tahap menginterpretasikan hasil *datamining* yang ditunjukkan pada tahap pemodelan dari fase *modeling*. Evaluasi dilaksanakan secara lebih mendetail dengan tujuan menyesuaikan model yang diperoleh agar sesuai dengan tujuan yang ingin dicapai dalam fase pertama.

6. *Deployment*

   Merupakan tahap penyebaran, di mana penyusunan laporan atau presentasi dari pengetahuan yang didapat dari evaluasi di proses *data mining* dilakukan.

2.6 Aplikasi Pendukung Pemodelan

2.6.1 *RapidMiner* Versi 5.3.015

RapidMiner merupakan *software* pengolahan *data mining* yang dapat digunakan pada lingkungan untuk *machine learning*, *data mining*, *text mining*, dan lain-lain. *RapidMiner* menyediakan prosedur *data mining* dan *machine learning* termasuk: ETL (*extraction, transformation, loading*), *data preprocessing*, *modelling*, dan evaluasi. Proses *data mining* merupakan susunan dari operator-operator yang *nestable*, dideskripsikan dengan XML, didukung GUI, juga ditulis dalam bahasa pemrograman Java. Berikut beberapa fungsi menu utama *RapidMiner* seperti :

1. *Process control*, merupakan fungsi yang diperuntukkan guna mengontrol aliran pemprosesan, seperti perulangan atau *conditional branch*.
2. *Utility*, digunakan pada pengelompokan sub *process*, juga *macro* dan *logger*.
3. *Repository access*, diperuntukkan dalam pembacaan dan penulisan *repository*.
4. *Import*, digunakan untuk membaca dari berbagai format *external*.
5. *Export*, digunakan untuk menulis data ke berbagai format *external*.
6. *Data Transformation*, berfungsi untuk mentransformasi data dan metadata.
7. *Modelling*, untuk proses *data mining* yang sesungguhnya. Seperti klasifikasi, regresi, *clustering*, aturan asosiasi, dan sebagainya.
8. *Evaluation*, untuk menghitung kualitas dari *modelling*.

2.6.2 *Microsoft Visual Basic* Versi 6.0

Secara singkat tentang *Microsoft Visual Basic* yang sering digunakan dalam dunia pemrograman merupakan sebuah bahasa pemrograman yang mana menyediakan *Integrated Development Environtment* (IDE) visual untuk merancang program perangkat lunak berbasis sistem operasi *Microsoft Windows* dengan berbasis pada model pemrograman (COM). Para *programmer* dapat membangun aplikasi bisnis dengan menggunakan komponen yang sudah ada di *Microsoft Visual Basic* itu sendiri, meski beberapa karakteristik obyek ada yang tidak dapat dilakukan pada *visual basic*. *Visual Basic* menjadi populer karena kemudahan desain *form* secara *visual* dan kemampuannya untuk menggunakan komponen yang dibuat pihak lain.