BAB 2
TINJAUAN PUSTAKA

2.1 Data Mining

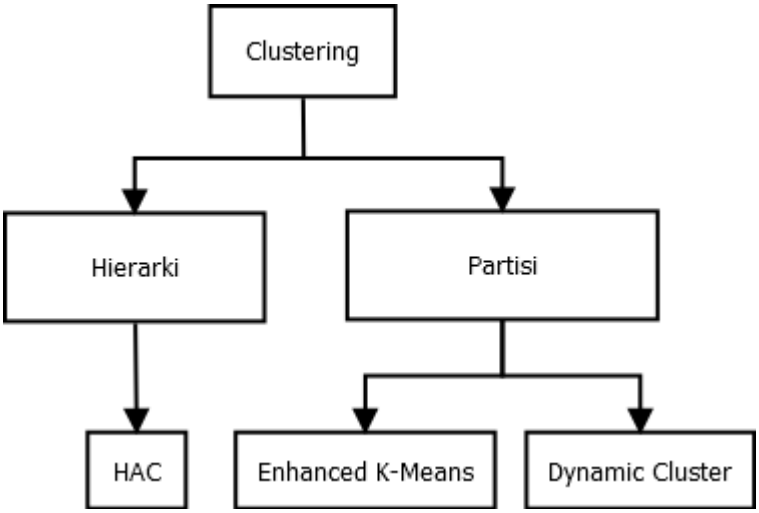
Data mining merupakan suatu metode pengolahan data yang digunakan untuk menemukan pola tersembunyi dari data yang akan diolah, sehingga akan menghasilkan informasi yang bermanfaat dan dapat digunakan untuk mengambil keputusan (Santosa B. , 2007). Dalam *data mining* terkadang ditemui data yang tidak memiliki label kelas sehingga data tersebut perlu dikelompokkan berdasarkan kemiripan dan ketidakmiripan antar data atau yang sering disebut dengan pembelajaran tidak terbimbing (*unsupervised learning*). Teknik dalam *data mining* yang tepat untuk menyelesaikan masalah tersebut adalah *clustering*, karena *clustering* tidak menggunakan label kelas pada data ketika akan memulai proses. Di akhir proses, *clustering* akan menghasilkan label kelas baru pada data yang telah diolah.

2.2 Clustering

Clustering merupakan metode yang digunakan untuk mencari dan mengelompokkan data berdasarkan kemiripan dan ketidakmiripan antar data. Hasil *clustering* dapat dikatakan baik jika data didalam suatu cluster memiliki kemiripan yang tinggi dan memiliki ketidakmiripan yang tinggi dengan data di cluster lain. Kualitas hasil cluster yang terbentuk bergantung pada karakteristik data dan algoritma yang digunakan, sehingga kesamaan karakteristik antar data dengan algoritma sangat diperlukan agar dapat menghasilkan hasil *clustering* yang baik.

2.3 Teknik-teknik Dalam Clustering

Clustering berdasarkan strukturnya terbagi menjadi dua yaitu hierarki dan partisi. Cabang teknik clustering dapat dilihat di Gambar 2.1



Gambar 2.1 Cabang Teknik Clustering

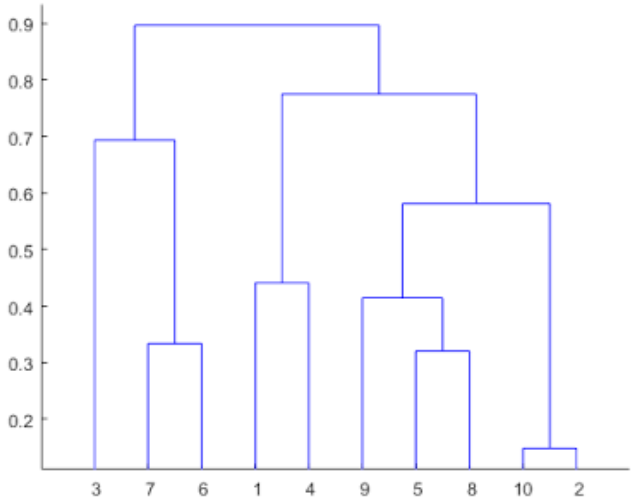
Hierarki adalah suatu metode pengelompokkan data yang dimulai dengan mengelompokkan dua atau lebih objek yang memiliki kesamaan paling dekat. Kemudian proses diteruskan ke objek lain yang memiliki kedekatan kedua. Demikian seterusnya sehingga *cluster* akan membentuk semacam pohon yang memperlihatkan ada hierarki yang jelas antar objek, dari yang paling mirip sampai yang paling tidak mirip. Secara logika semua objek pada akhirnya hanya akan membentuk sebuah *cluster* (Santosa S. , 2010). Salah satu algoritma hierarki adalah Hierarchical Agglomerative Clustering (HAC).

Berbeda dengan metode hierarki, metode partisi memulai proses dengan menentukan terlebih dahulu perkiraan jumlah *cluster* yang diinginkan (dua *cluster*, tiga *cluster*, atau lain sebagainya). Setelah jumlah *cluster* diketahui, baru proses *clustering* dilakukan tanpa mengikuti proses hierarki. Metode ini biasa disebut dengan K-Means *clustering* (Santosa S. , 2010). Algoritma yang termasuk kedalam kelompok partisi adalah Enhanced K-Means dan Dynamic Cluster.

2.3.1 Algoritma Hierarchical Agglomerative Clustering (HAC)

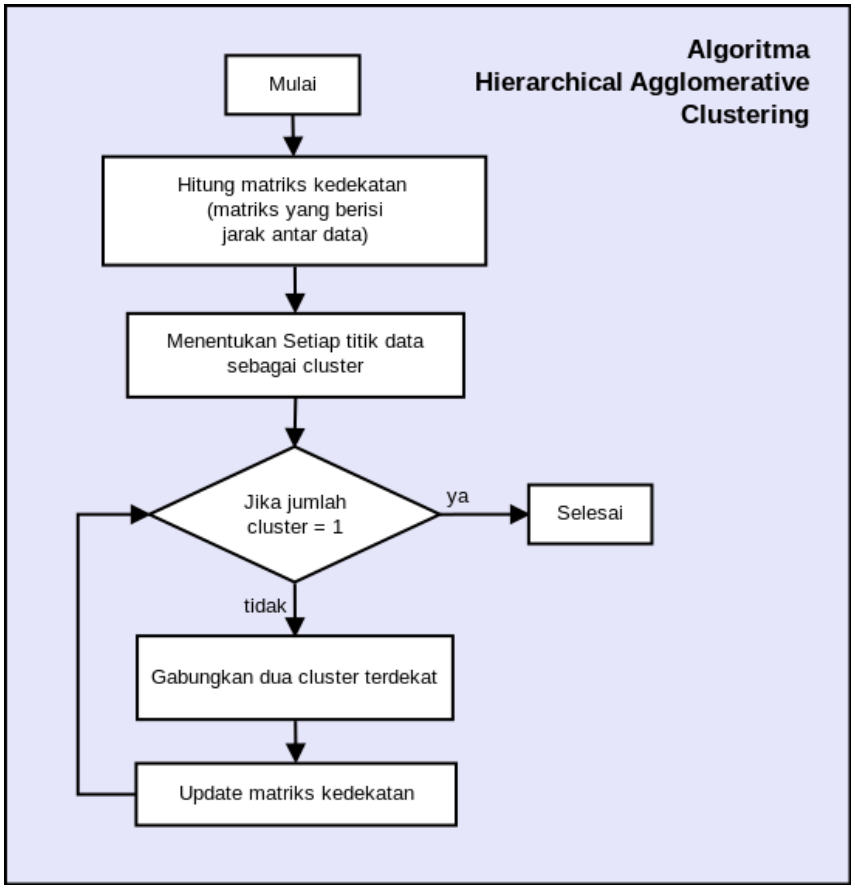
Hierarchical Agglomerative Clustering (HAC) merupakan metode *clustering* berbasis hierarki dengan pendekatan *bottom-up*, yaitu proses *clustering* dimulai dari masing-masing data sebagai satu buah *cluster*, kemudian secara rekursif mencari *cluster* terdekat sebagai pasangan untuk bergabung sebagai satu *cluster*

yang lebih besar. Proses tersebut diulang terus sehingga tampak bergerak ke atas membentuk hierarki (Prasetyo, 2014). *Clustering* berbasis hierarki sering ditampilkan dalam bentuk grafis menggunakan diagram yang mirip *tree* (pohon) yang disebut dengan dendogram. Dendogram menampilkan hubungan *cluster* dengan *subcluster*-nya dalam urutan *cluster* yang digabung. Contoh bentuk dendogram dapat dilihat di Gambar 2.2.



Gambar 2.2 Bentuk Dendogram

Pada Gambar 2.2, sumbu x merupakan titik data dan sumbu y merupakan jarak antar data. Metode ini dimulai dengan membentuk *cluster* pada setiap obyek. Kemudian dua obyek yang memiliki jarak terdekat digabungkan. Selanjutnya obyek ketiga digabung dengan obyek lain yang memiliki jarak terdekat dan membentuk *cluster* baru. Proses akan berlanjut hingga akhirnya terbentuk satu *cluster* yang terdiri dari keseluruhan obyek. *Flowchart* algoritma HAC dapat dilihat di Gambar 2.3.

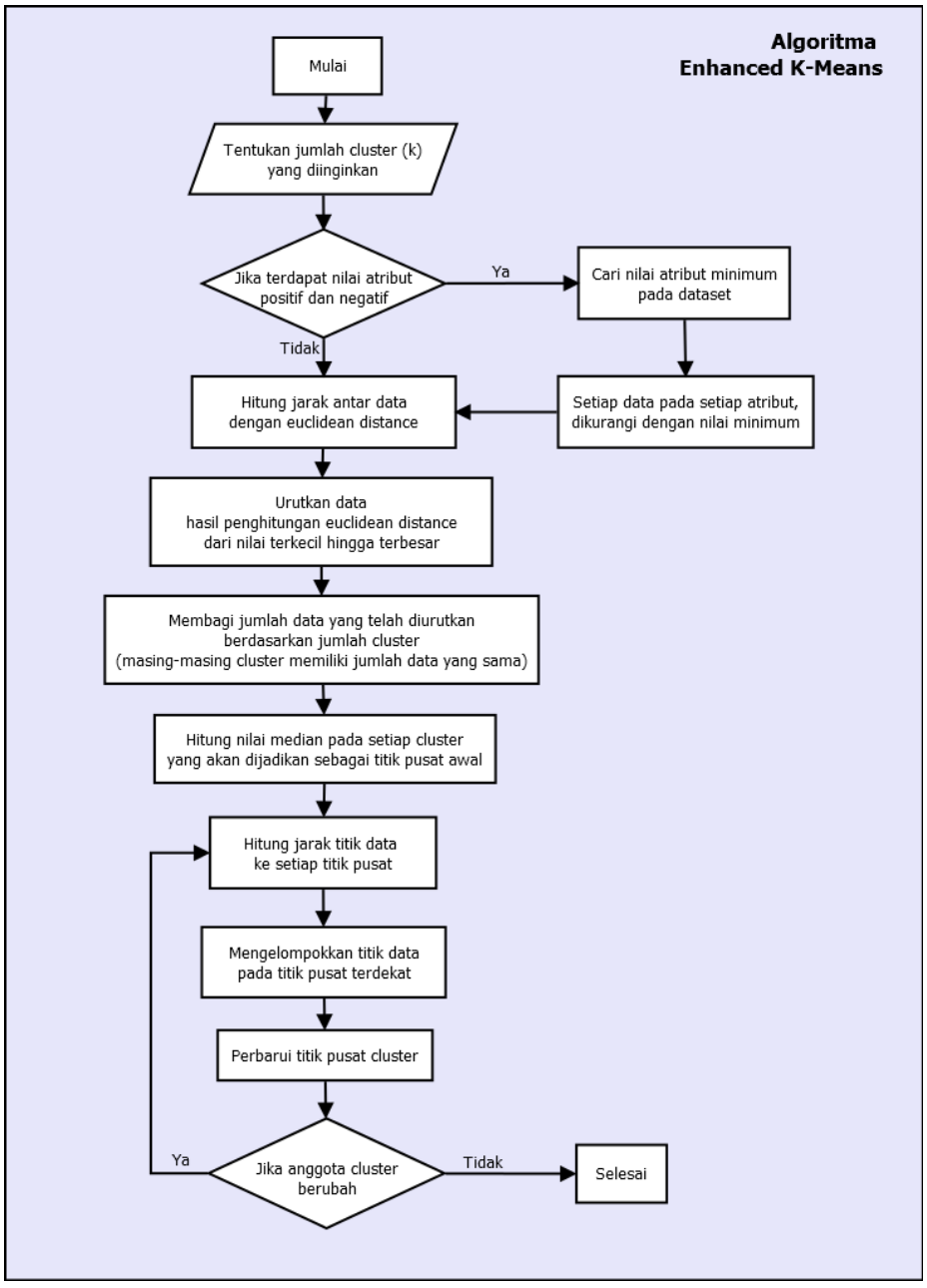


Gambar 2.3 Flowchart Hierarchical Agglomerative Clustering

Pada Gambar 2.3 terlihat bahwa algoritma Hierarchical Agglomerative Clustering memulai proses dengan menghitung matriks kedekatan kemudian menentukan setiap titik data sebagai cluster, jika jumlah cluster = 1 maka proses selesai, jika tidak maka proses dilanjutkan pada penggabungan dua cluster terdekat, selanjutnya melakukan pembaruan matriks kedekatan dan proses diulang kembali pada kondisi jumlah cluster = 1, hingga ditemukan cluster berjumlah 1.

2.3.2 Enhanced K-Means

Enhanced K-Means (EKM) merupakan algoritma yang dikembangkan dari K-Means oleh Madu Yedla, dkk (Madu, Srinivasa, & Pathakota, 2010). Algoritma ini dikembangkan untuk menutupi sensitifitas penentuan titik pusat awal pada K-Means. Berikut ini proses dari algoritma Enhanced K-Means yang diperlihatkan pada Gambar 2.4.



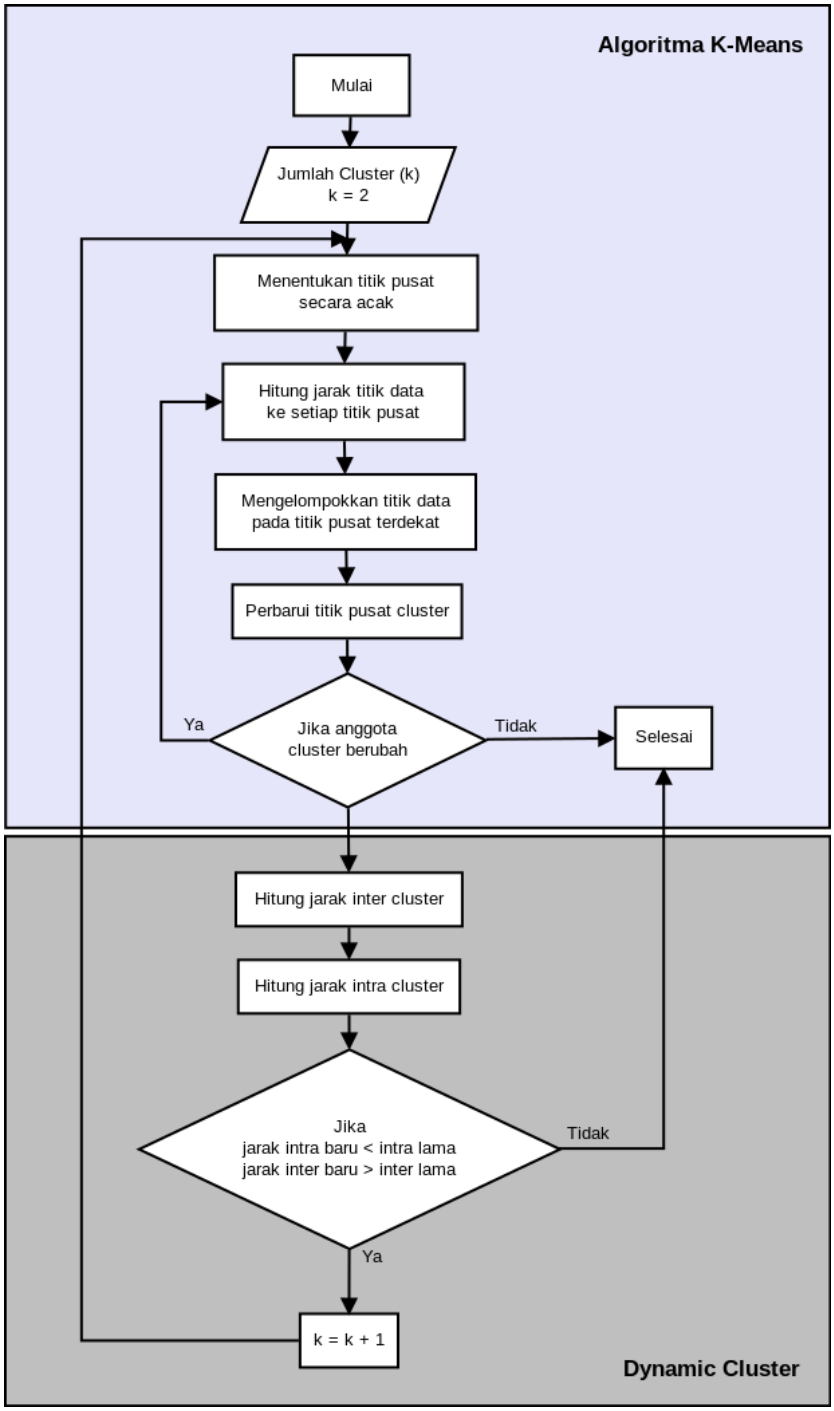
Gambar 2.4 Flowchart Enhanced K-Means

Pada Gambar 2.4 terlihat bahwa algoritma Enhanced K-Means memulai proses dengan menentukan jumlah *cluster* terlebih dahulu secara manual, setelah itu pada *dataset* jika terdapat nilai atribut positif dan negatif maka cari nilai atribut *minimum* pada *dataset* kemudian setiap data pada setiap atribut dikurangi nilai *minimum,* selanjutnya menghitung jarak antar setiap titik data ke titik data yang lain, sedangkan jika nilai atribut hanya positif atau negatif, maka langsung dilakukan penghitungan jarak antar setiap titik data ke titik data yang lain menggunakan *euclidean distance*, hasil dari penghitungan tersebut kemudian akan diurutkan dari nilai terkecil hingga terbesar, kemudian kelompokkan urutan tersebut berdasarkan jumlah *cluster* yang telah ditentukan dengan jumlah data yang sama untuk setiap cluster. Selanjutnya menghitung nilai median pada setiap cluster yang akan dijadikan sebagai titik pusat awal. Setelah titik pusat didapatkan langkah selanjutnya sama seperti pada algoritma k-means yaitu menghitung titik data terhadap setiap titik pusat dan mengelompokkan setiap titik data ke titik pusat terdekat, proses ini dilakukan berulang hingga tidak ditemukan lagi titik data yang berpindah *cluster*.

Kelebihan dari algoritma Enhanced K-Means adalah mampu menentukan titik pusat awal yang lebih baik dibandingkan dengan algoritma K-Means terutama pada data dengan jumlah yang banyak. Akan tetapi algoritma Enhanced K-Means masih mengadopsi sensitifitas penentuan perkiraan awal jumlah *cluster* pada algoritma K-Means, sehingga tidak dapat mengetahui jumlah *cluster* optimal pada dataset yang akan diolah.

2.3.3 Dynamic Cluster

Dynamic Cluster (DKM) adalah algoritma yang dikembangkan untuk melakukan optimasi pada K-Means, berbeda dengan Enhanced K-Means, Dynamic Cluster menambahkan algoritmanya untuk menutupi sensitifitas penentuan dugaan jumlah *cluster* pada K-Means, sehingga algoritma ini dapat menemukan jumlah *cluster* yang optimal. Berikut ini proses dari algoritma Dynamic Cluster yang diperlihatkan pada Gambar 2.5.



Gambar 2.5 Flowchart Implementasi Dynamic Cluster Pada K-Means

Dari Gambar 2.5 terlihat bahwa algoritma Dynamic Cluster bekerja setelah proses K-Means, kemudian algoritma Dynamic Cluster akan menghitung jarak inter dan jarak intra dari proses sebelumnya, jika jarak intra baru lebih kecil dari jarak intra lama dan jarak inter baru lebih besar dari jarak inter lama maka

algoritma Dynamic Cluster akan menambahkan satu pada jumlah *cluster* sebelumnya, proses ini akan dilakukan berulang hingga algoritma Dynamic Cluster menemukan jarak intra yang sekecil mungkin dan jarak inter yang sebesar mungkin.

Kelebihan dari algoritma Dynamic Cluster adalah mampu menentukan jumlah cluster yang lebih baik dibandingkan dengan algoritma K-Means. Akan tetapi algoritma Dynamic Cluster masih mengadopsi sensitifitas penentuan titik pusat awal pada algoritma K-Means, sehingga tidak dapat mengetahui jumlah *cluster* terbaik.

2.3.3.1 Jarak Intra

Istilah intra digunakan untuk mengukur kekompakan dari suatu kelompok (Widiarina & Romi, 2015). Dengan kata lain jarak intra adalah jarak antar titik data dengan titik pusat *(centroid)* dalam suatu kelompok atau *cluster*, sehingga semakin kecil jarak intra maka akan semakin baik hasil *cluster* yang dihasilkan. Berikut ini gambaran mengenai intra yang diperlihatkan pada Gambar 2.6.



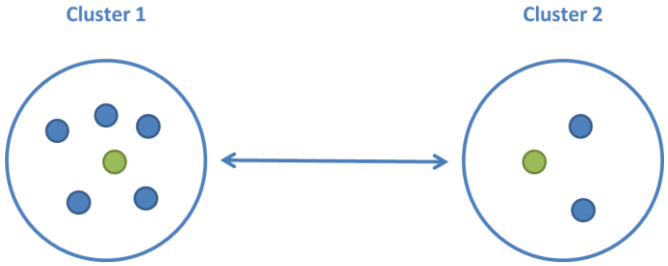
Gambar 2.6 Jarak Intra

Berdasarkan Gambar 2.6, jarak titik data berwarna biru ke titik pusat berwarna hijau diistilahkan sebagai intra.

2.3.3.2 Jarak Inter

Istilah inter adalah minimum jarak antar pusat *cluster*, inter digunakan untuk mengukur pemisahan antar *cluster* (Widiarina & Romi, 2015). Dari keterangan tersebut maka disimpulkan bahwa jarak inter adalah jarak antar pusat *cluster* dengan pusat *cluster* lainnya, sehingga semakin besar jarak inter maka akan

semakin baik hasil *cluster* yang dihasilkan. Berikut ini gambaran mengenai intra yang diperlihatkan pada Gambar 2.7.



Gambar 2.7 Jarak Inter

Berdasarkan Gambar 2.7, jarak antar titik pusat berwarna hijau diistilahkan sebagai inter.

2.4 Evaluasi Hasil *Clustering*

Untuk mengetahui hasil suatu *cluster* optimal atau tidak maka perlu dilakukan pengukuran atau evaluasi pada hasil *cluster* yang terbentuk sehingga dapat diketahui apakah hasil *cluster* yang terbentuk optimal atau tidak. Teknik pengevaluasian hasil *cluster* yang umum digunakan oleh para peneliti di bidang *clustering* adalah Davies-Bouldin Index (DBI).

2.4.1 Davies-Bouldin Index

Davies-Bouldin Index merupakan salah satu metode yang bertujuan untuk mengevaluasi *cluster* pada suatu metode *clustering* berdasarkan nilai inter dan intra yang diperkenalkan oleh David L. Davies dan Donald W. Bouldin.

Sum of Square Within Cluster (SSW) pada pengukuran DBI merupakan persamaan yang digunakan untuk mengetahui jarak intra yang dapat dihitung menggunakan persamaan sebagai berikut :

$$SSW_i = \frac{1}{m}\sum_{j=1}^{m_j} d(x_j, c_i) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.1)$$

Sedangkan Sum of Square Between Cluster (SSB) merupakan persamaan yang digunakan untuk mengetahui jarak inter yang dapat dihitung menggunakan persamaaan sebagai berikut :

$$SSB_{i,j} = d(c_i, c_j) \text{ ..... (2.2)}$$

Setelah diperoleh nilai jarak inter dan jarak intra maka akan dilakukan pengukuran rasio (R_{ij}) untuk menghasilkan nilai perbandingan antara *cluster* ke-i dan cluster ke-j. Kriteria cluster yang baik adalah *cluster* yang memiliki nilai intra sekecil mungkin dan nilai inter sebesar mungkin. Nilai rasio dapat dihitung menggunakan persamaan berikut ini :

$$R_{i,j} = \frac{SSW_i + SSW_j}{SSB_{i,j}} \text{ ..... (2.3)}$$

Nilai rasio yang dihasilkan dari persamaan tersebut akan digunakan untuk mencari nilai DBI menggunakan persamaan berikut :

$$DBI = \frac{1}{K}\sum_{i=1}^{K} \max(R_{i,j}) \text{ ..... (2.4)}$$

Dari perhitungan tersebut akan dihasilkan nilai DBI, semakin kecil nilai DBI atau semakin mendekati angka 0 maka semakin baik *cluster* yang diperoleh (Devi & Geetharamani, 2012).

Keterangan :

m_j = Merupakan jumlah data yang berbeda dalam cluster ke-*i*

c_i = Merupakan titik pusat cluster ke-*i*

k = Merupakan jumlah cluster

2.5 Euclidean Distance

Untuk menghitung jarak antar titik data pada *clustering* digunakan penghitungan *Euclidean Distance*. *Euclidean Distance* merupakan salah satu fungsi yang digunakan dalam pengukuran similaritas antar data dalam *clustering*. Fungsi ini menyatakan jarak diantara dua data poin. *Euclidean Distance* menentukan akar kuadrat perbedaan atau jarak diantara dua buah koordinat (Irani, Pise, & Phatak, 2016).

Euclidean distance diantara dua data poin $X = (x_1, x_2, \dots, x_m)$ dan $Y = (y_1, y_2, \dots, y_m)$ diformulasikan oleh persamaan sebagai berikut:

$$d(X,Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_m - y_m)^2} \quad \text{……...…} \quad (2.5)$$

Keterangan :

X = Koordinat titik data ke-1

Y = Koordinat titik data ke-2