BAB II

TINJAUAN PUSTAKA

2.1 Analisis Deskriptif

Analisis Deskriptif Menurut Hasan (2001), statistik deskriptif atau statistik deduktif adalah bagian dari statistik mempelajari cara pengumpulan data dan penyajian data sehingga mudah dipahami. Statistik deskriptif hanya berhubungan dengan hal menguraikan atau memberikan keterangan-keterangan mengenai suatu data atau keadaan atau fenomena. Dengan kata lain statistik deskriptif berfungsi menerangkan keadaan, gejala, atau persoalan (Nasution, 2017).

2.2 *Knowledge Discovery in Database* (KDD)

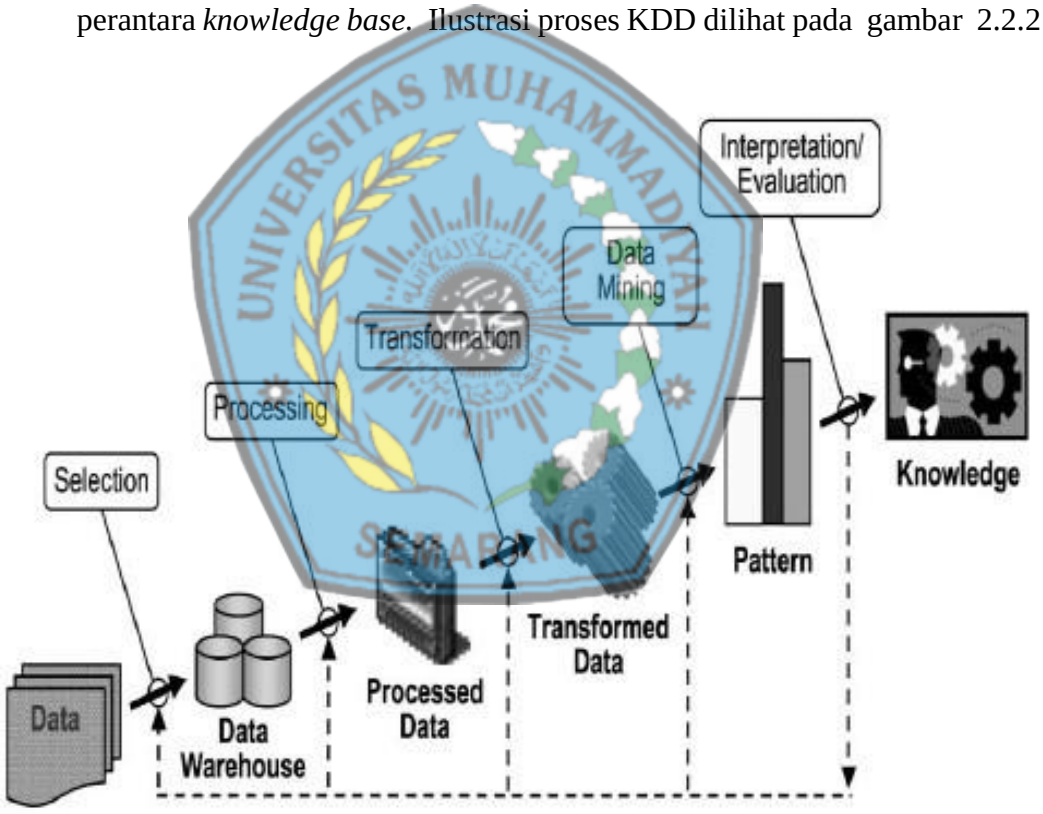
2.2.1 Definisi *Knowledge Discovery in Database* (KDD)

Knowledge Discovery in Database (KDD) adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola, atau hubungan dalam set data berukuran besar. Penambangan data (*data maining*) merupakan bagian dari *Knowledge Discovery in Database* yang merupakan kegiatan yang meliputi pengumpulan, pemakain data historis untuk menemukan keteraturan, pola, atau hubungan yang berukuran besar (Santosa, 2007).

http://repository.unimus.ac.id

2.2.2 Tahapan *Knowledge Discovery in Database* (KDD)

Menurut Han and Kamber (2006), penambangan data tidak dapat dipisahkan dari proses *Knowledge Discovery in Database* (KDD). KDD merupakan sebuah proses mengubah data menjadi suatu informasi yang berguna. Tahapan KDD merupakan suatu ringkasan proses penambangan data (*data maining*) dan dapat dibagi menjadi beberapa tahap. Tahap – tahap tersebut bersifat interaktif di mana pemakai terlibat langsung atau melalui perantara *knowledge base*. Ilustrasi proses KDD dilihat pada gambar 2.2.2



Gambar 2.2.2 Proses *Knowledge Discovery in Database* (Jejaring, 2019)

http://repository.unimus.ac.id

Tahapan proses *Knowledge Discovery in Database* (KDD) adalah sebagai berikut:

1. **Data *Selection***
    - Menciptakan himpunan data target , pemilihan himpunan data, atau memfokuskan pada subset variabel atau sampel data, dimana penemuan (discovery) akan dilakukan.
    - Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining, disimpan dalam suatu berkas, terpisah dari basis data operasional.
2. ***Pre-processing/ Cleaning***
    - Pemprosesan pendahuluan dan pembersihan data merupakan operasi dasar seperti penghapusan *noise* dilakukan.
    - Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD.
    - Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang *inkonsisten*, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi).
    - Dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

http://repository.unimus.ac.id

3. ***Transformation***

   - Pencarian fitur-fitur yang berguna untuk mempresentasikan data bergantung kepada goal yang ingin dicapai.
   - Merupakan proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining. Proses ini merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data

4. **Data mining**

   - Pemilihan tugas data mining, pemilihan goal dari proses KDD misalnya klasifikasi, *regresi, clustering*, dll.
   - Pemilihan algoritma *data mining* untuk pencarian (*searching*)
   - Proses *Data mining* yaitu proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan.

5. **Interpretation/ Evaluation**

   - Penerjemahan pola-pola yang dihasilkan dari *data mining*.
   - Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan.

http://repository.unimus.ac.id

- Tahap ini merupakan bagian dari proses KDD yang mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

Tahap 1 sampai dengan tahap 4 merupakan berbagai bentuk dari data *preprocessing,* dimana data disiapkan untuk dilakukan penambangan data (*data mining*). *Data mining* hanyalah salah satu langkah dari keseluruhan proses dalam *Knowledge Discovery in Database* (KDD). (Han & Kamber, 2006)

2.3 Penambangan Data (*Data Mining)*

2.3.1 Definisi Penambangan Data (*Data Mining*)

Data mining adalah suatu proses penambangan informasi penting dari suatu data. Informasi penting ini didapat dari suatu proses yang amat rumit seperti menggunakan *artificial intelligence*, teknik statistik, ilmu matematika, *machine learning*, dan lain sebagainya. Teknik-teknik rumit tersebut nantinya akan mengidentifikasi dan mengekstraksi informasi yang bermanfaat dari suatu database besar. (Efraim Turban, dkk 2005).

Data mining adalah serangkaian proses untuk menggali nilai tambahan dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual (Pramudiono, 2006).

2.3.2 Teknik Penambangan Data (*Data Mining*)

Teknik dalam penambangan data adalah sebagai berikut (Hermawati, 2013):

a. *Classification* (Klasifikasi)

http://repository.unimus.ac.id

Klasifikasi adalah menentukan sebuah *record* data baru ke salah satu dari beberapa kategori (atau kelas) yang telah didefinisikan sebelumnya. Disebut juga dengan *'supervised learning'.*

b. *Clustering* (Klasterisasi)

Klasterisasi adalah mempartisi data-set menjadi beberapa sub-set atau kelompok sedemikian rupa sehingga elemen – elemen dari suatu kelompok tertentu memiliki *set property* yang di*share* bersama, dengan tingkat *similaritas* yang tinggi dalam satu kelompok dan tingkat *similaritas* antar kelompok yang rendah. Disebut juga dengan '*unsupervised learning'.*

c. *Association Rule Discovery* (Kaidah asosiasi)

Mendeteksi kumpulan atribut – atribut yang muncul bersamaan (*co-occur*) dalam frekuensi yang sering, dan membentuk sejumlah kaidah dari kumpulan – kumpulan tersebut.

2.4 Standarisasi Data / Pembakuan Data

Menurut Hair,et al. (2016) pembekuan data adalah proses mengkonversi nilai masing – masing variabel awal menjadi nilai standar dengan rata – rata 0 dan standar deviasi 1 untuk menghilangkan bias yang disebabkan karena perbedaaan skala dari beberapa variabel yang digunakan dalam analisis.

2.5 Clustering

2.5.1 Definisi *Clustering*

Clustering yaitu menemukan kumpulan obyek hingga obyek-obyek dalam satu kelompok sama (atau punya hubungan) dengan yang lain dan

http://repository.unimus.ac.id

berbeda (atau tidak berhubungan) dengan obyek – obyek dalam kelompok lain. Tujuan dari *clustering* adalah untuk meminimalkan jarak di dalam *cluster* dan memaksimalkan jarak antar *cluster.*

Dalam mengukur jarak dalam *clustering* dapat dilakukan dengan menggunakan *Euclidean Distance. Euclidean Distance* merupakan pengukuran jarak obyek dan pusat *cluster* yang banyak digunakan secara luas dalam berbagai kasus *pattern matching,* termasuk *clustering.* (Astri Widiastuti Setiyawati, 2017) *Euclidean Distance* dinyatakan dengan persamaan :

$$Dist = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2} \qquad (2.1)$$

Dimana :

n = jumlah fitur dalam suatu data

k = indeks data

Pk = nilai atribut (fitur) ke-k dari p

qk = nilai atribut (fitur) ke-k dari q

2.5.2 Tipe Clustering

Clustering merupakan suatu kumpulan dari keseluruhan cluster. Beberapa tipe penting dari *clustering* adalah sebagai berikut (Hermawati, 2013):

1. *Partitional vs Hierarchical*

http://repository.unimus.ac.id

Partitional clustering adalah pembagian objek data kedalam sub himpunan (*cluster*) yang tidak *overlap* sedemikian hingga tiap objek data berada dalam tepat satu sub-himpunan.

Hierarchical clustering merupakan sebuah himpunan *cluster* bersarang yang diatur sebagai suatu pohon *hirarki*.Tiap simpul (*cluster*) dalam pohon (kecuali simpul daun) merupakan gabungan dari anaknya (sub*cluster*) dan simpul akar berisi semua objek

2. *Exclusive vs non-exclusive*

   Semua bentuk clustering merupakan *exclusive clustering* karena setiap objek berada tepat pada satu cluster. Sebaliknya dalam *overlapping* atau *non-exclusive clustering*, sebuah objek dapat berada di lebih dari satu cluster secara bersamaan.

3. *Fuzzy vs non-Fuzzy*

   Fuzzy clustering, sebuah titik termasuk dalam setiap cluster dengan suatu nilai bobot antara 0 dan 1. Jumlah dari bobot - bobot tersebut sama dengan 1. *Clustering probabilitas* mempunyai karakteristik yang sama.

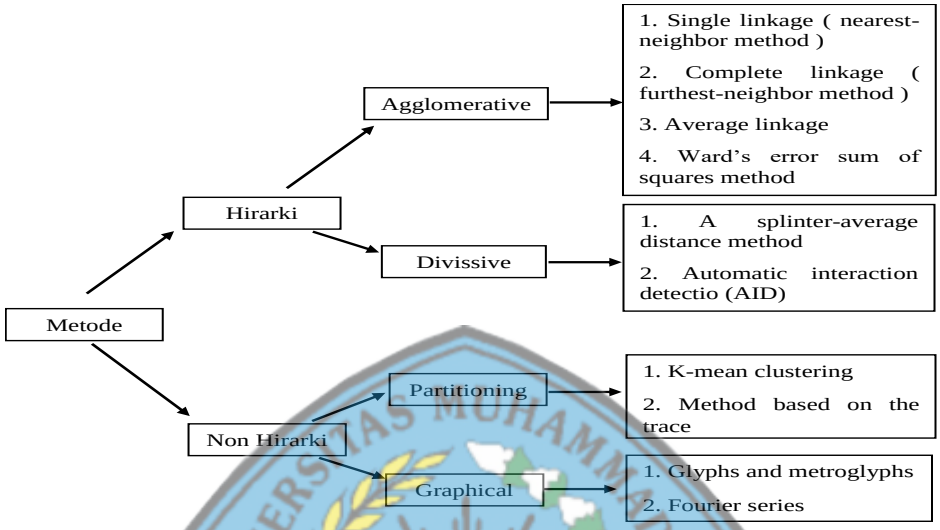
4. *Partial vs Complete*

   Complete clustering, setiap objek ditempatkan dalam sebuah *cluster*. Tetapi dalam *partial clustering*, tidak semua objek ditempatkan dalam sebuah *cluster*. Kemungkinan ada objek yang tidak tepat untuk ditempatkan di salah satu *cluster*, misalkan berupa *outlier* atau *noise*.

http://repository.unimus.ac.id

2.5.3 Metode Clustering

Metode dalam membuat cluster ada banyak sekali, seperti yang digambarkan dalam diagram di bawah ini (Anwar Hidayat, 2014) :



Gambar 2.3 *Diagram Analisis Cluster*

2.5.4 *Hirarchial Methode*

Metode ini memulai pengelompokan dengan dua atau lebih objek yang mempunyai kesamaan paling dekat. Kemudian proses diteruskan ke objek lain yang mempunyai kedekatan kedua. Demikian seterusnya sehingga cluster akan membentuk semacam "pohon", di mana ada hirarki (tingkatan) yang jelas antar objek, dari yang paling mirip sampai paling tidak mirip. Secara logika semua objek pada akhirnya akan membentuk sebuah cluster. *Dendogram* biasanya digunakan untuk membantu memperjelas proses hirarki tersebut. Metode hirarki cluster terdapat dua tipe dasar yaitu *agglomerative (pemusatan)* dan *divisive (penyebaran)*. Dalam metode *agglomerative*, setiap obyek atau observasi dianggap sebagai sebuah cluster tersendiri. Dalam tahap selanjutnya, dua *cluster* yang mempunyai kemiripan digabungkan menjadi sebuah *cluster* baru demikian seterusnya. Sebaliknya, dalam

http://repository.unimus.ac.id

metode *divisive* kita beranjak dari sebuah cluster besar yang terdiri dari semua obyek atau observasi. Selanjutnya, obyek atau observasi yang paling tinggi nilai ketidakmiripannya kita pisahkan demikian seterusnya (Anwar Hidayat, 2014).

1. Agglomerative

Agglomerative ada lima metode yang cukup terkenal, yaitu: *Single Linkage, Complete Linkage, Average Linkage, Ward's Method, Centroid Method* (Anwar Hidayat, 2014).

- *Single Linkage*, prosedur ini didasarkan pada jarak terkecil. Jika dua obyek terpisah oleh jarak yang pendek maka kedua obyek tersebut akan digabung menjadi satu cluster dan demikian saterusnya.
- *Complete Linkage,* berlawanan dengan *Single Linkage* prosedur ini pengelompokkannya berdasarkan jarak terjauh.
- *Average Linkage,* prosedure ini hampir sama dengan Single Linkage maupun *Complete Linkage,* namun kriteria yang digunakan adalah rata-rata jarak seluruh individu dalam suatu cluster dengan jarak seluruh individu dalam cluster yang lain.
- *Ward's Method*, jarak antara dua cluster dalam metode ini berdasarkan *total sum of square* dua cluster pada masing-masing variabel.
- *Centroid Method*, jarak antara dua cluster dalam metode ini berdasarkan jarak *centroid* dua cluster yang bersangkutan.

http://repository.unimus.ac.id

2.5.5 *Non-Hirarchial Methode*

Berbeda dengan metode *hirarki*, metode ini justru dimulai dengan terlebih dahulu menentukan jumlah *cluster* yang diinginkan (dua cluster, tiga cluster atau yang lain). Setelah jumlah cluster diketahui, baru proses cluster dilakukan tanpa mengikuti proses hirarki. Metode ini biasa disebut dengan *K-Means Cluster*. Metode *non hirarki* tidak meliputi proses "*treelike construction*". Justru menempatkan objek-objek ke dalam cluster sekaligus sehingga terbentuk sejumlah cluster tertentu. Langkah pertama adalah memilih sebuah cluster sebagai inisial cluster pusat, dan semua objek dalam jarak tertentu ditempatkan pada cluster yang terbentuk. Kemudian memilih cluster selanjutnya dan penempatan dilanjutkan sampai semua objek ditempatkan. Objek-objek bisa ditempatkan lagi jika jaraknya lebih dekat pada *cluster* lain daripada cluster asalnya.

Metode *non hirarki cluster* berkaitan dengan *K-means custering*, dan ada tiga pendekatan yang digunakan untuk menempatkan masing-masing observasi pada satu cluster.

1. *Sequential Threshold*, Metode *Sequential Threshold*

Sequential Threshold, Metode *Sequential Threshold* memulai dengan pemilihan satu cluster dan menempatkan semua objek yang berada pada jarak tertentu ke dalamnya. Jika semua objek yang berada pada jarak tertentu telah dimasukkan, kemudian cluster yang kedua dipilih dan menempatkan semua objek yang berjarak tertentu ke dalamnya. Kemudian cluster ketiga dipilih dan proses dilanjutkan seperti yang sebelumnya.

http://repository.unimus.ac.id

2. *Parallel Threshold*, Metode *Parallel Threshold*

Parallel Threshold, Metode *Parallel Threshold* merupakan kebalikan dari pendekatan yang pertama yaitu dengan memilih sejumlah cluster secara bersamaan dan menempatkan objek-objek kedalam cluster yang memiliki jarak antar muka terdekat. Pada saat proses berlangsung, jarak antar muka dapat ditentukan untuk memasukkan beberapa objek ke dalam cluster-cluster. Juga beberapa variasi pada metode ini, yaitu sisa objek-objek tidak dikelompokkan jika berada di luar jarak tertentu dari sejumlah cluster.

3. *Optimization*

Optimization, Metode ketiga adalah serupa dengan kedua metode sebelumnya kecuali bahwa metode ini memungkinkan untuk menempatkan kembali objek-objek ke dalam cluster yang lebih dekat.

2.6 *Partitioning Around Medoids* (PAM)

Partitioning Around Medoids (PAM) atau di kenal dengan K – Medoids adalah algoritma pengelompokan yang berkaitan dengan algoritma K – Means dan algoritma K – Medoids.

Algoritma Partitioning Around Medoids (PAM) dikembangkan oleh Leonard Kuaufman dan Peter J. Rousseuw. Algoritma ini sangat mirip dengan algoritma K – Means, terutama karena kedua algoritma ini partitional. Dengan kata lain, kedua algoritma ini memecah dataset menjadi kelompok –kelompok dan kedua algoritma ini berusaha untuk meminimalkan kesalahan. Tetapi

http://repository.unimus.ac.id

algoritma *Partitioning Around Medoids* (PAM) berkerja dengan menggunakan Medoids, yang merupakan entitas dari dataset yang mewakili kelompok dimana ia dimasukkan (Astri Widiastuti Setiyawati, 2017).

Algoritma Partitioning Around Medoids (PAM) menggunakan metode patsisi *clustering* untuk mengelompokkan sekumpulan n obyek menjadi sejumlah cluster. Alogaritma ini menggunakan obyek pada kumpulan obyek untuk mewakili sebuah *cluster*. Obyek yang terpilih untuk mewakili sebuah *cluster* disebut dengan medoids. *Cluster dibangun dengan menghitung kedekatan yang dimiliki antara medoids dengan obyek* non-medoid.

2.6.1 *Algoritma Partitioning Around Medoids* (PAM)

Algoritma Partitioning Around Medoids (PAM) atau K – Medoids adalah sebagai berikut (Han & Kamber, 2006):

1. Secara acak pilih *k* obyek pada sekumpulan *n* obyek sebagai *medoids.*
2. Ulangi langkah 3 hingga langkah 6.
3. Tempatkan obyek *non-medoids* ke dalam *cluster* yang paling dekat dengan medoids.
4. Secara acak pilih 0_{random} sebuah obyek *non-medoids.*
5. Hitung total biaya, *S,* dari pertukaran *medoids* 0_j dengan 0_{random}.
6. Jika S < 0 maka tukar 0_j dengan 0_{random} untuk membentuk sekumpulan *k* obyek baru sebagai medoids.
7. Hingga tidak ada perubahan.

Nilai total biaya/*cost* dinyatakan dengan persamaan:

http://repository.unimus.ac.id

$$\text{Total } cost = \sum Dist \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2} \quad (2.2)$$

Nilai S dinyatakan dengan persamaan:

S = Total *cost* baru – Total *cost* lama

Dimana :

Total *cost* baru = jumlah biaya/*cost* non-medoids.

Total *cost* lama = jumlah biaya/*cost* medoids.

n = jumlah fitur dalam suatu data

k = indeks data

Pk = nilai atribut (fitur) ke-k dari p

qk = nilai atribut (fitur) ke-k dari q

K – medoids sangat mirip dengan K – means, perbedaan utama diantara dua alogaritma tersebut adalah jika pada K – means *cluster* diwakili dengan pusat dari *cluster*, sedangkan pada K – medoids *cluster* diwakili oleh obyek terdekat dari pusat *cluster*.

2.7 *Silhoutte*

2.7.1 *Silhotte Index* (SI)

Jika DBI digunakan untuk mengukur validasi seluruh *cluster* dalam set data, maka *Silhouette Index* (SI) dapat digunakan untuk menvalidasi baik sebuah data, *cluster* tunggal (satu *cluster* dari sejumlah *cluster*), atau bahkan keseluruhan *cluster*. Metode ini yang paling banyak digunakan untuk menvalidasi *cluster* yang menggabungkan nilai kohesi dan separasi.

http://repository.unimus.ac.id

Untuk menghitung nilai SI dari sebuah data ke-i, ada 2 komponen yaitu a_i dan b_i. a_i adalah rata – rata jarak ke-i terhadap semua data lainnya dalam satu *cluster* sedangkan b_i didapatkan dengan menghitung rata – rata jarak data ke-i terhadap semua data dari *cluster* yang lain tidak dalam satu *cluster* dengan data ke-i, kemudian diambil yang terkecil (Tan et al, 2006 & Petrovic,2003).

$$a_i^j = \frac{1}{m_j - 1} \sum_{\substack{r=1 \\ r \neq 1}}^{m_j} d\,(x_i^j, x_r^j),\ i = 1,2, \ldots . m_j \qquad (2.3)$$

Dimana :

j = *cluster*

i = *index* data

a_i^j = rata – rata jarak data ke-i terhadap semua data lainnya dalam satu *cluster.*

m_j = jumlah data dalam *cluster* ke-j.

$d\,(x_i^j, x_r^j)$ adalah jarak data ke-i dengan data ke-r dalam satu *cluster j.*

Berikut formula untuk menghitung b_i^j :

$$b_i^j = \min_{\substack{n = 1, \ldots, k \\ n \neq j}} \left\{ \frac{1}{m_n} \sum_{\substack{r=1 \\ r \neq 1}}^{m_n} d\,(x_i^j, x_r^n) \right\} i = 1,2, \ldots, m_j \qquad (2.4)$$

Dimana :

j = *cluster*

http://repository.unimus.ac.id

n = *cluster*

i = *index* data

m_n = banyak data dalam satu *cluster*

b_i^j = nilai terkecil dari rata – rata jarak data ke-i terhadap semua data dari *cluster* yang lain tidak dalam satu *cluster* dengan data ke-i

$d\ (x_i^j, x_r^n)$ adalah jarak data ke-i dalam satu *cluster j* dengan data ke-r dalam suatu *cluster* n

Untuk mendapatkan *Silhouette Index* (SI) data ke-i menggunakan persamaan berikut :

$$SI_i^j = \frac{b_i^j - a_i^j}{\max(b_i^j, a_i^j)} \qquad (2.5)$$

Dimana :

SI_i^j = *Silhouette Index* data ke-i dalam satu *cluster*

b_i^j = nilai terkecil dari data rata – rata jarak data ke-i terhadap semua data dari *cluster* yang lain tidak dalam satu *cluster* dengan data ke-i

a_i^j = rata – rata jarak data ke-i terhadap semua data lainnya dalam satu *cluster*

Nilai a_i mengukur seberapa tidak mirip sebuah data dengan *cluster* yang diikutinya, nilai yang semakin kecil menandakan semakin tepatnya data tersebut berada dalam *cluster* tersebut. Nilai b_i yang besar menandakan seberapa jeleknya

http://repository.unimus.ac.id

data terhadap *cluster* yang lain. Nilai SI yang didapat dalam rentang [-1,+1]. Nilai SI yang mendekati 1 menandakan bahwa data tersebut semakin tepat berada dalam *cluster* tersebut. Nilai SI negatif ($a_i > b_i$ menandakan bahwa data tersebut tidak tepat berada di dalam *cluster* tersebut karena lebih dekat ke *cluster* yang lain). SI bernilai 0 (atau mendekati 0) berarti data tersebut posisinya berada di perbatasan di antara dua *cluster*.

Untuk nilai SI dari sebuah *cluster* didapatkan dengan menghitung rata – rata nilai SI semua data yang bergabung dalam *cluster* tersebut, seperti pada persamaan berikut :

$$SI_j = \frac{1}{m_j} \sum_{i=1}^{m_j} SI_i^j \qquad (2.6)$$

Dimana :

SI_j= rata – rata *Silhouette Index cluster* j

m_j = jumlah data dalam *cluster* ke-j

SI_i^j = *Silhouette Index* data ke-i dalam satu *cluster*

i = *index*

Sementara nilai SI global didapatkan dengan menghitung rata – rata nilai SI dari semua *cluster* seperti pada persamaan berikut :

$$SI = \frac{1}{k} \sum_{j=1}^{k} SI_j \qquad (2.7)$$

http://repository.unimus.ac.id

Dimana :

SI = rata – rata *Silhouette Index* dari *database*

k = jumlah *cluster*

SI_j = rata – rata *Silhouette Index cluster* j

2.7.2 *Silhouette Coeffocient* (SC)

Silhouette Coeffocient adalah suatu metode yang digunakan untuk mengetahui apakah *cluster* yang terbentuk adalah *cluster* yang memiliki struktur kuat, struktur baik, struktur lemah, maupun struktur yang buruk. Untuk menghitung nilai *Silhouette Coeffocient*, terlebih dahulu menghitung *Silhouette Index* dari sebuah data ke-i. Nilai *Silhouette Coeffocient* didapatkan dengan mencari nilai maksimal dari *Silhouette Index* Global dari jumlah *cluster* 2 sampai jumlah *cluster* n-1, seperti persamaan berikut:

$$SC = maks_k\ SI\ (k) \qquad (2.8)$$

Dimana :

SC = *Silhouette Coeffocient*

SI = *Silhouette Index*

k = jumlah *cluster*

Kriteria subjektif pengukuran baik atau tidaknya pengelompokan berdasarkan *Silhouette Coeffocient* (SC) menurut Kauffman dan Roesseeuw (1990) dapat dilihat tabel 2.7.2 dibawah.

http://repository.unimus.ac.id

Tabel 2.7.2 Kriteria subjektif pengukuran baik atau tidaknya pengelompokan berdasarkan *Silhouette Coeffocient* (SC)

Nilai SC	Interpretasi Kauffman dan Raoesseeuw
0.71 – 1.00	Struktur Kuat
0.51 – 0.70	Struktur Baik
0.26 – 0.50	Struktur Lemah
00.25	Struktur Buruk

http://repository.unimus.ac.id