BAB II

LANDASAN TEORI

2.1. *Knowledge Discovery in Database* (KDD)

Menurut Putra (2017), KDD merupakan suatu pola kegiatan *non-trivial* (tidak biasa) guna untuk mencari serta mengidentifikasi pola (*pattern*) yang terjadi pada sekumpulan data, dimana pola yang ditemukan bersifat sah (yang sebenarnya) baru dapat bermanfaat dan dapat dimengerti. Adapun KDD berhubungan dengan hal-hal yang bersifat teknik integraso dan penemuan ilmiah, interpretasi dan visualisasi dari pola-pola sejumlah kumpulan data.

Adapun tahap dari KDD:

1. *Data Selection*

    a. Membuat kumpulan atau himpunan dari *data target*, pemilihan kumpulan data dengan pola tertentu, atau memfokuskan pada variabel atau data sampel, dimana proses kegiatan aktivitas penemuan (*discovery*) dilakukan.

    b. Pemilihan/seleksi data-data dari sekumpulan data yang memiliki keterkaitan hubungan secara operasional perlu dilakukan terlebih dahulu yang kemudian baru dapat melanjutkan ke tahap penggalian informasi yang ada ke dalam proses KDD. Kumpulan atau himpunan data hasil dari proses seleksi yang akan digunakan pada proses *data*

mining disimpan dalam suatu tempat/berkas yang terpisah dari kumpulan basis data operasional yang menjadi sumber data.

2. *Pre-processing*/*Cleaning*

   a. *Pre-processing* merupakan operasi yang paling dasar dilakukan seperti menghilangkan *noise*. Barulah proses *data mining* dapat dilaksanakan, perlu dilakukan terlebih dahulu proses *cleaning* pada kumpulan *data set* yang menjadi *data set* awal pada proses KDD.

   b. Proses *cleaning* meliputi kegiatan antara lain membuang data duplikat serta memeriksa data yang tidak konsisten atau relevan, kemudian memperbaiki kesalahan pada data tersebut.

   c. Dilakukan proses *enrichment*/memperkaya data set dengan data lain atau informasi lain (eksternal).

3. *Transformation*

   a. Pencarian pola-pola pada data serta hubungan antar fitur yang terdapat pada data/keterkaitan data.

   b. Keterkaitan *attribute* atau fitur yang dipakai untuk menampilkan data bergantung kepada hasil yang ingin diperoleh.

   c. Merupakan proses perubahan/penggabungan (transformasi) pada *data set* yang telah dipilih.

4. *Data Mining*

   a. Melakukan proses pemilihan tugas serta pemilihan hasil dari proses KDD misalnya klasifikasi, regresi, *clustering*, dll.

   b. Pemilihan metode algoritma *data mining* yang akan dipakai.

   c. Melakukan proses *data mining* yaitu proses mencari pola-pola atau informasi yang menarik dalam data set yang terpilih dengan menggunakan teknik serta metode tertentu.

5. *Interpretation*/*Evaluation*

   a. Penerjemahan pola/*pattern* yang terjadi atau pola yang dihasilkan dari proses *data mining*.

   b. Pola/*pattern* informasi yang dihasilkan dari proses *data mining* akan ditampilkan dalam format/bentuk yang mudah dimengerti oleh pihak yang berkepentingan

2.2. Metode *Clustering*

Menurut Freitas (2014), *clustering* terdiri dari pembagian data yang telah di-*mining*, ke dalam beberapa kelompok (atau *cluster*) *instance data*, secara demikian rupa sehingga: (a) setiap *cluster* memiliki *instance* yang sangat mirip (*similar*, atau "dekat"), dan (b) *instance* di setiap *cluster* sangat berbeda (atau "jauh sekali") dari *instance* di *cluster* yang lain. Algoritma *clustering* memaksimalkan kesamaan *intra-cluster* (atau *within-cluster*).

2.3. Algoritma *X-Means Clustering*

Menurut Nguyen, Kowalczyk, Orłowski, & Ziółkowski (2016), *X-Means* merupakan perluasan dari algoritma *K-Means clustering*. *X-Means* dapat mengidentifikasi jumlah *cluster* terbaik dengan berdasarkan *Bayesian Information Criterion* (BIC). Algoritma *clustering* ini membutuhkan pengaturan *k cluster* yang lebih fleksibel daripada *K-Means*, yaitu perlu menentukan nilai maksimal *max_k* dan nilai minimal *min_k* dari *k cluster*. *X-Means* kemudian akan mengidentifikasi nilai *k* di kisaran nilai minimal dan maksimal [*min_k*, *max_k*] yang harus dipilih.

Dua langkah dalam *X-Means*:

1. *Improve-Params*

   Menjalankan *K-Means* hingga konvergen.

2. *Improve-Structure*

   Memutuskan apakah *cluster* harus dibagi menjadi dua *sub-cluster* atau tidak berdasarkan BIC.

Algoritma berhenti ketika jumlah *cluster* mencapai jumlah maksimum *cluster max_k* yang telah ditetapkan di awal.

2.4. *Bayesian Information Criterion* (BIC)

Menurut Konishi & Kitagawa (2010), *Bayesian Information Criterion* (BIC) merupakan kriteria evaluasi untuk model yang didefinisikan berdasarkan

probabilitas posteriornya. Formula untuk BIC (Chen, Zhu, Hu, & Principe, 2013) adalah:

$$BIC = -2 \log L_{max} + k \log_n$$

Rumus 2.1. *Bayesian Information Criterion* (BIC)

Dimana:

n = jumlah data yang diamati (ukuran sampel)

L_{max} = *maximum likelihood estimation*

k = jumlah *cluster*

2.5. *Davies-Bouldin Index* (DBI)

Menurut Citra dan Geetharamani (2012), *Davies-Bouldin Index* merupakan salah satu metode evaluasi internal yang mengukur evaluasi *cluster* pada suatu metode pengelompokan yang didasarkan pada nilai kohesi dan separasi. Dalam suatu pengelompokan, kohesi didefinisikan sebagai jumlah dari kedekatan data terhadap *centroid* dari *cluster* yang diikuti. Sedangkan separasi didasarkan pada jarak antar *centroid* dari *cluster*-nya.

Sum of Square within Cluster (SSW) merupakan persamaan yang digunakan untuk mengetahui matrik kohesi dalam sebuah *cluster* ke-*i* yang dirumuskan sebagai berikut :

$$SSW_i = \frac{1}{m_i} \sum_{j=i}^{m_i} d(x_j, c_i)$$

Rumus 2.2. *Sum of Square within Cluster* (SSW)

Dimana:

m_i = jumlah data dalam *cluster* ke-*i*

c_i = *centroid cluster* ke-*i*

d() = jarak setiap data ke *centroid*

x_j = atribut data ke-*j*

Sum of Square between Cluster (SSB) merupakan persamaan yang digunakan untuk mengetahui separasi antar *cluster* yang dihitung menggunakan persamaan:

$$SSB_{i,j} = d(c_i, c_j)$$

Rumus 2.3. *Sum of Square between Cluster* (SSB)

Dimana:

d() = jarak *centroid cluster* ke-*i* dengan *centroid cluster* ke-*j*

Setelah nilai kohesi dan separasi diperoleh, kemudian dilakukan pengukuran rasio (R_{ij}) untuk mengetahui nilai perbandingan antara *cluster* ke-*i* dan *cluster* ke-*j*. *Cluster* yang baik adalah *cluster* yang memiliki nilai kohesi sekecil mungkin dan separasi yang sebesar mungkin. Nilai rasio dihitung menggunakan persamaan sebagai berikut:

$$R_{i,j} = \frac{SSW_i + SSW_j}{SSB_{i,j}}$$

Rumus 2.4. Pengukuran Nilai Rasio

Dimana:

SSW_i = *Sum of Square within Cluster* ke-*i*

SSW_j = *Sum of Square within Cluster* ke-*j*

$SSB_{i,j}$ = *Sum of Square between Cluster* ke-*i* dan *cluster* ke-*j*

Nilai rasio yang diperoleh tersebut digunakan untuk mencari nilai *Davies-Bouldin Index* (DBI) dari persamaan berikut:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j}(R_{i,j})$$

Rumus 2.5. *Davies-Bouldin Index*

Dimana:

k = jumlah *cluster* yang digunakan

$R_{i,j}$ = rasio antara *cluster* ke-*i* dan *cluster* ke-*j*

Semakin kecil nilai DBI yang diperoleh (non-negatif >= 0), maka semakin baik *cluster* yang diperoleh dari pengelompokan *K-Means* yang digunakan.

2.6. Algoritma *K-Means Clustering*

K-means adalah algoritma *clustering* parsial sederhana berbasis *prototype* yang mencoba menemukan *cluster k* yang tidak tumpang tindih (*non-overlapping*). *Cluster* ini diwakili oleh *centroids*. Sebuah *centroid* dari sebuah *cluster* biasanya adalah *mean* dari titik-titik di *cluster* tersebut (Wu, 2014).

Menurut Setiawan (2016), langkah-langkah melakukan *clustering* dengan metode *K-Means* adalah sebagai berikut:

1. Pilih jumlah *cluster k.*

2. Inisialisasi *k* pusat *cluster* ini bisa dilakukan dengan berbagai cara. Namun yang paling sering dilakukan adalah dengan cara *random*. Pusat-pusat *cluster* diberi nilai awal dengan angka-angka *random*.

3. Alokasikan semua data/objek ke *cluster* terdekat, kedekatan dua objek ditentukan berdasarkan jarak kedua objek tersebut. Demikian juga kedekatan suatu data ke *cluster* tertentu ditentukan jarak antara data dengan pusat *cluster*. Jarak antara satu data dengan satu *cluster* tertentu akan menentukan suatu data masuk dalam *cluster* mana. Untuk menghitung jarak semua data ke setiap titik pusat *cluster* dapat menggunakan teori jarak *Euclidean* yang dirumuskan sebagai berikut:

$$D(i,j) = \sqrt{(X_{1i} - X_{1j})^2 + (X_{2i} - X_{2j})^2 + \cdots + (X_{ki} - X_{kj})^2}$$

Rumus 2.5. *Euclidean*

Dimana:

$D_{(i,j)}$ = Jarak data ke *i* ke pusat *cluster j*

X_{ki} = Data ke *i* atribut data ke *k*

X_{kj} = Titik pusat ke *j* pada atribut ke *k*

4. Hitung kembali pusat *cluster* dengan keanggotaan *cluster* yang sekarang. Pusat *cluster* adalah rata-rata dari semua data/objek dalam *cluster* tertentu. Jika dikehendaki bisa juga menggunakan median dari *cluster* tersebut. Jadi rata-rata (*mean*) bukan satu-satunya ukuran yang bisa dipakai.

5. Tugaskan lagi setiap objek memakai pusat *cluster* yang baru. Jika pusat *cluster* tidak berubah lagi makan proses *clustering* selesai. Atau, kembali ke langkah nomor 3 sampai pusat *cluster* tidak berubah lagi.

2.7. *Principal Compenent Analysis*

Principal Component Analysis (PCA) merupakan teknik yang biasa digunakan untuk menyederhanakan suatu data, dengan cara mentransformasi data secara linier sehingga terbentuk sistem koordinat baru dengan varians maksimum. Analisis PCA dapat digunakan untuk mereduksi dimensi suatu data tanpa mengurangi karakteristik data tersebut secara signifikan (Rahayu & Mustakim, 2017).

2.8. Visualisasi Data

Visualisasi data adalah tampilan informasi dalam format grafik atau tabel. Tujuan visualisasi adalah representasi dari informasi yang disampaikan kepada pihak-pihak yang melihat agar mudah memahami informasi yang disampaikan tersebut (Wahyudi, 2013).

2.8.1. *Slicer*

Slicer merupakan bentuk penyaring (*filter*) interaktif yang ada di dalam Microsoft Power BI, untuk memilih satu atau lebih elemen yang ingin disaring di dalam sebuah *report*. Dalam sebuah *report*, dapat ditambahkan beberapa *slicer* yang berbeda, sehingga data di dalam sebuah

report dapat tersaring dengan berbagai macam kriteria. *Slicer* biasanya berbasis *text*, tetapi bisa juga berbentuk grafik sederhana (Aspin, 2016).

2.8.2. *Clustered Column Chart*

Clustered Column Chart digunakan untuk membandingkan semua kategori dan sub-kategorinya sebagai bagian dari sebuah kesatuan, sehingga dapat melihat dalam setiap kategori, sub-kategori mana yang memiliki nilai yang lebih tinggi jika dibandingkan dengan yang lainnya (Rad, 2017).

2.8.3. *Table*

Table merupakan salah satu cara paling berguna untuk melihat data, terutama jika perlu melihat dan mencari informasi secara detail (Clark, 2017).

2.9. *Dashboard*

Dashboard adalah sebuah tampilan visual dari informasi terpenting yang dibutuhkan untuk mencapai satu atau lebih tujuan, digabungkan dan diatur pada sebuah layar, menjadi informasi yang dibutuhkan dan dapat dilihat secara sekilas. Tampilan visual disini mengandung pengertian bahwa penyajian informasi harus dirancang sebaik mungkin, sehingga mata manusia dapat menangkap informasi secara cepat dan otak manusia dapat memahami maknanya secara benar. *Dashboard* ditampilkan pada satu monitor komputer penuh, yang bersifat kritis, agar kira dapat melihatnya dengan cepat, sehingga dengan melihat *dashboard* saja, kita dapat mengetahui hal-hal yang perlu diketahui (Rohayati, 2014).

2.10. RapidMiner

RapidMiner merupakan *software* yang menyediakan sebuah *integrated environment* untuk semua langkah proses *data mining*, *graphical user interface* (GUI) yang mudah digunakan untuk perancangan proses *data mining*, visualisasi data dan hasil, validasi dan optimalisasi proses *data mining*, serta untuk penyebaran secara otomatis dan memungkinkan integrasi ke dalam sistem yang lebih kompleks (Hofmann & Klinkenberg, 2016).

2.11. Microsoft Power BI

Microsoft Power BI merupakan sebuah *software* yang memberikan kemampuan untuk menganalisis dan menyajikan data, dan membentuk dan memberikan hasil dengan mudah dan impresif. Power BI juga memberikan layanan untuk menyimpan dan membagikan data dalam bentuk *dashboard* dan *reports* (Aspin, 2016).