BAB II

LANDASAN TEORI

2.1 Sistem Pendukung Keputusan

Michael S. Scoot Morton pada awal tahun 1970-an memperkenalkan konsep Sistem Pendukung Keputusan (SPK). Konsep SPK ditandai dengan sistem interaktif yang berbasis komputer yang membantu pengambilan keputusan dengan memanfaatkan data dan model yang bertujuan untuk menyelesaikan suatu masalah yang bersifat tidak terstruktur dan semi terstruktur (Meliala,2012). SPK merupakan sebatas alat bantu untuk pembuatan suatu keputusan.

Proses-proses yang terjadi pada kerangka kerja SPK dibedakan sebagai berikut (Beerawa, 2012).

1. Terstruktur

   Mengacu pada permasalahan rutin dan terulang untuk solusi standar yang ada.

2. Tidak Terstruktur

   Merupakan permasalahan kompleks dimana tidak ada solusi yang serta-merta dapat diselesaikan. Terdapat tiga fase proses yang menjadikan suatu proses menjadi tidak struktur, yaitu

   a. *Intelligence,* merupakan pencarian kondisi-kondisi yang dapat menghasilkan keputusan
   b. *Design,* menemukan, mengembangkan, dan menganalisis materi-materi yang memungkinkan untuk dikerjakan.

c. *Choice* , merupakan pemilihan dari materi-materi yang tersedia, materi mana yang akan dikerjakan.

3. Semi terstruktur

   Sistem Pendukung Keputusan dalam mencapai tujuannya harus memiliki tiga syarat, yaitu sederhana, mudah dikontrol, mudah beradaptasi, lengkap pada hal-hal penting, dan mudah digunakan atau berkomunikasi dengan sistem.

2.1.1 Komponen Sistem Pendukung Keputusan

Sistem Pendukung keputusan terdiri dari beberapa kompnen-komponen sebagai berikut (Meliala, 2012).

1. Manajemen Data

   Mencakup *database* yang mengandung data yang relevan dan diatur oleh sistem yang disebut *Database Management System (DMBS).*

2. Manajemen Model

   Merupakan paket perangkat lunak yang memasukkan model-model finansial, statistik, ilmu manajemen, ataupun model kuantitatif yang menyediakan kemampuan analisis sistem dan *management software.*

3. Antarmuka Pengguna

   Merupakan media interaksi antara sistem dengan pengguna, sehingga pengguna dapat berkomunikasi dengan mudah dengan sistem yang telah dibuat.

4. Subsistem Berbasis Pengetahuan

   Subsistem yang dapat mendukung subsistem lain atau bertindak sebagai komponen yang dapat berdiri sendiri.

2.1.2 Proses Pengambilan Keputusan

Keempat tahapan proses pengambilan keputusan dapat dijelaskan sebagai berikut (Meliala, 2012).

1. Tahap penelurusan (*Intelligence)*

   Tahap tempat dilakukannya pendefinisian masalah serta indentifikasi informasi yang dibutuhkan yang berkaitan dengan persoalan yang dihadapi.

2. Perancangan (*Design)*

   Tahap tempat dilakukannya analisa dalam kaitan mencari atau merumuskan alternatif pemecahan masalah. Di dalam tahapan ini juga terdapat proses merancang atau membangun model pemecahan masalah dan disusun menjadi berbagai alternatif pemecahan masalah.

3. Pemilihan (*Choice)*

   Setelah perancangan alternatif solusi, maka dipilih kembali alternatif solusi yang paling sesuai. Kemudahan tahapan ini dipengaruhi oleh suatu alternatif memiliki nilai kuantitas tertentu dan terukur.

4. Implementasi (*Implementation)*

   Merupakan tahapan terakhir dari keputusan yang telah diambil. Pada tahapan ini juga memerlukan susunan serangkaian tindakan yang terencana agar hasil keputusan dapat dipantau dan disesuaikan bila terjadi suatu perbaikan.

2.2 Data Mining

Data mining merupakan sebuah proses yang digunakan untuk menemukan suatu hubungan, pola, dan tren baru melalui penyaringan data yang sangat besar yang disimpan di dalam tempat penyimpanan dan menggunakan teknik pengenalan pola seperti teknik statistika dan teknik matematika (Larose, 2005). *Data mining*

juga disebut sebagai *Knowledge Discovery in Database* (KDD). KDD merupakan kegiatan yang meliputi pengumpulan, pemakaian data, dan historis untuk menemukan suatu keteraturan pola dan hubungan antar data dalam ukuran yang besar (Santoso, 2007). Pola-pola yang ditemukan dalam data harus mempunyai arti dan pola tersebut harus memberikan keuntungan (Witten, 2005).

2.2.1 Karakteristik Data Mining

Data mining memiliki beberapa karakteristik, karakteristik tersebut adalah sebagai berikut (Davies, 2004).

1. *Data mining* merupakan penemuan sesuatu yang tersembunyi dan mempunyai pola data tertentu yang tidak diketahui sebelumnya.
2. *Data mining* menggunakan data yang besar untuk mendapatkan hasil yang dapat dipercaya.
3. *Data mining* bertujuan untuk membuat keputusan yang kritis, terutama dalam hal bisnis.

2.2.2 Pengolahan Data Mining

Data mining terdiri dari beberapa metode pengolahan, yaitu (Larose, 2005).

1. Predictive modelling

   Predictive modelling merupakan pengolahan *data mining* dengan melakukan prediksi atau peramalan. Metode ini bertujuan untuk membangun suatu model prediksi nilai yang mempunyai ciri khas tertentu. Algoritma yang menggunakan metode ini adalah Linear Regression, Neural Network, Support Vector Machine, dan lain-lain.

2. Association

   Association merupakan teknik *data mining* yang mempelajari hubungan antar data. Contoh penggunaan teknik ini seperti menganalisis perilaku mahasiswa yang terlambat mengumpulkan tugas. Contoh algoritma yang menggunakan teknik ini adalah FP-Growth dan Apriori.

3. Clustering

   Clustering merupakan teknik pengelompokkan data ke dalam kelompok tertentu. Contoh algoritma yang menggunakan teknik *clustering* adalah K-Means dan Fuzzy C-Means. Contoh kasus untuk *clustering* adalah terdapat empat fakultas di dalam Universitas Multimedia Nusantara, di dalam masing-masing fakultas dikelompokkan kembali menjadi beberapa program studi.

4. Classification

   Classification merupakan teknik mengklasifikasikan data. Perbedaan dengan *clustering* adalah terletak pada data yang digunakan. Pada *clustering* tidak terdapatnya variabel dependen, sedangkan pada *classification* diwajibkan terdapat variabel dependen. Contoh algortima yang menggunakan metode *classification* adalah ID3 dan K Nearest Neighbors.

2.3 Decision Tree (Pohon Keputusan)

Decision tree atau pohon keputusan merupakan metode yang mengubah fakta yang sangat besar menjadi pohon keputusan yang mempresentasikan aturan. Aturan tersebut dapat dengan mudah dipahami dan dapat diekspresikan dalam bentuk bahasa basis data seperti *Structured Query Language* untuk mencari *record* pada suatu kategori tertentu. Selain itu, pohon keputusan berguna untuk

mengeksplorasi data dan menemukan hubungan yang tersembunyi antara sejumlah calon variabel input dengan sejumlah variabel target (Kusrini, 2008).

Pohon keputusan dipilih karena proses pembangunannya lebih cepat dan hasil dari model yang dibangun lebih mudah untuk dipahami sehingga *decision tree* merupakan metode klasifikasi yang paling popular untuk digunakan (Ginting, 2014).

2.4 Algoritma C4.5

Algoritma C4.5 merupakan bagian dari *decision tree*, dua model algoritma tersebut tidak bisa dipisahkan, karena untuk membangun sebuah *decision tree* dibutuhkan algoritma C4.5 (Andriani, 2013). Algoritma C4.5 merupakan pengembangan dari algoritma ID3 (*Iterative Dichotomiser)* yang dikembangkan oleh J. Ross Quinlan seorang peneliti di bidang mesin pembelajaran. Algoritma C4.5 mengalami perkembangan dengan berbasis *supervised learning* (Han & Kamber, 2001).

Perbaikan C4.5 mengalami puncaknya karena adanya perbaikan yang meliputi metode untuk menangani *numeric attributes, missing values, noisy* data, dan aturan yang menghasilkan rules dan *trees* (Witten dkk, 2005).

Beberapa tahapan dalam membuat sebuah *decision tree* dalama algoritma C4.5 (Larose, 2005), yaitu:

1. Mempersiapkan data *training.* Data *training* didapat dari data *history* yang pernah terjadi sebelumnya dan sudah dikelompokkan dalam kelas-kelas tertentu.
2. Menghitung akar dari pohon. Akar akan diambil dengan cara menghitung nilai *gain* dari masing-masing atribut. Nilai *gain* yang paling tinggi akan menjadi akar

yang pertama. Sebelum menghitung nilai *gain* dari atribut, terlebih dahulu hitung nilai *entropy.*

$$\text{Entropy (S)} = \sum_{i=1}^{n} -\text{Pi}\log_2\text{Pi} \quad \ldots(2.1)$$

Keterengan :

S = himpunan kasus

N = jumlah partisi S

P_i = proporsi S_i terhadap S

3. Kemudian hitung nilai *gain* dengan menggunakan rumus:

$$\text{Gain(S,A)} = \text{Entropy(S)} - \sum_{i=1}^{n} \left|\frac{S_i}{S}\right| * \text{Entropy (Si)} \quad \ldots(2.2)$$

Keterangan :

S = himpunan kasus

A = fitur

n = jumlah partisi atribut A

$|S_i|$ = proporsi Si terhadap S

|S| = jumlah kasus dalam S

4. Ulangi langkah ke dua dan langkah ke tiga sampai *record* terpartisi.

5. Proses *decision tree* akan berhenti ketika :

   a. Semua *record* dalam simpul n mendapat kelas yang sama

   b. Tidak ada atribut di dalam *record* yang dipartisi lagi

   c. Tidak ada *record* di dalam cabang yang kosong

Pada algoritma C4.5 untuk memperbaiki informasi dari *gain* dapat menggunakan *gain ratio.*

$$\text{Gain Ratio(S,A)} = \frac{\text{Gain (S,A)}}{\text{SplitInfo(S,A)}} \quad \ldots(2.3)$$

Dimana S merupakan ruang (*sample)* data yang digunakan untuk *training* dan A adalah atribut. *Gain(S,A)* merupakan informasi *gain* pada atribut A, *SplitInfo* adalah nilai *split information* pada atribut yang didapat dari rumus sebagai berikut.

$$\text{SplitInfo(S,A)} = -\sum_{i=1}^{n} \frac{\text{Si}}{\text{S}} log_2 \frac{S_i}{\text{S}} \quad \text{... (2.4)}$$

Entropy merupakan nilai yang digunakan untuk mengukur homogenitas suatu data yang mengandung kecocokan. (Linyu, dkk. 2001). *Gain* atau *Information Gain* didefinisikan sebagai perbedaan diantara informasi asli yang dibutuhkan dengan jumlah informasi baru yang didapatkan dari suatu partisi (Miswaningsih, 2015). *Split Information* didefinisikan sebagai *entropy* atau informasi yang potensial yang akan digunakan dalam rumus mencari *Gain Ratio* (Defiyanti dan Pardede, 2014). *Gain ratio* didefinisikan sebagai suatu cara untuk menguari bias terhadap atribut yang dipilih yang mempunyai nilai yang besar (Max, 2007).

2.5 Evaluasi

Untuk melakukan evaluasi maka dapat dilakukan dengan melakukan pengujian *confusion matrix* atau *cross validation* (Arisona, 2015).

2.5.1 Confusion Matrix

Confusion Matrix memberikan keputusan yang diperoleh dalam *testing* dan juga memberikan penilaian *performance* klasifikasi berdasarkan objek dengan benar maupun salah. Selain itu *confusion matrix* berisi informasi aktual dan prediksi.

Tabel 2.1 Confusion Matrix Tabel (Sun dkk., 2006).

		Predicted class			
		C_1	C_2	.....	C_k
True class	C_1	n_{11}	n_{12}	.....	n_{1k}
	C_2	n_{21}	n_{22}	.....	n_{2k}
	.	.	.	.	.
	.	.	.	.	.
	.	.	.	.	.
	C_k	n_{k1}	n_{k2}	.....	n_{kk}

True Class melambangkan data yang bersifat benar atau asli dan *Predicted Class* melambangkan data yang diprediksikan. N11, N22, …, Nkk merupakan data yang bernilai benar dimana nilai dari *true class* dan *predicted class* adalah sama.

$$\text{Accuracy} = \frac{N11 + N22 + N33 + \cdots + Nkk}{N11 + N12 + N13 + .. + Nkk} \quad \text{... (2.5)}$$

2.5.2 Cross Validation

Cross Validation merupakan salah satu teknik untuk melakukan validasi keakuratan sebuah model yang dibangun berdasarkan data set tertentu. Pembuatan model bertujuan untuk melakukan prediksi maupun klasifikasi terhadap suatu data baru yang boleh jadi belum pernah muncul di dalam data set. Data yang digunakan dalam pembangunan model tersebut adalah data *training* atau data latih, sedangkan data yang akan digunakan untuk memvalidasi model disebut sebagai data *testing* atau data uji (Arisona, 2015).

Dalam teknik pengujian data dengan menggunakan *cross validation*, terdapat dua jenis metode, metode tersebut adalah sebagai berikut (Arisona, 2015).

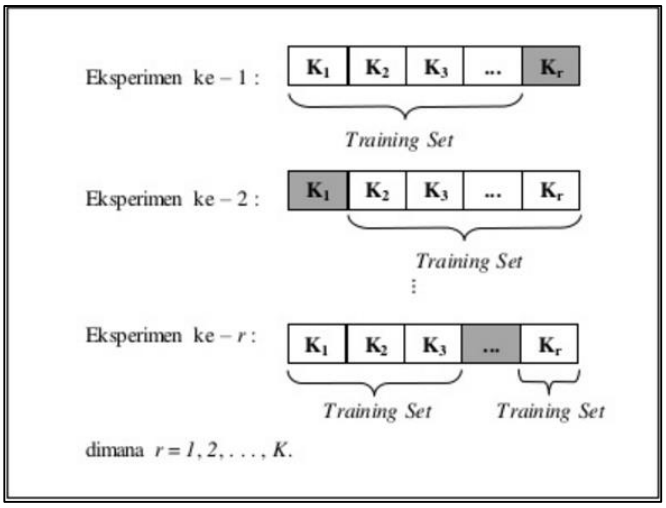
1. Metode Two-Fold Cross Validation

   Dalam metode ini, data digunakan beberapa kali dalam jumlah yang sama untuk data *training* dan data *testing*. Metode ini membagi data menjadi dua bagian, yaitu data *training* dan data *testing*, kemudian pada percobaan selanjutnya terjadi

pertukaran pada data *training* menjadi data *testing*, dan data *testing* menjadi data *training.* Total *erorr* didapat dengan menjumlahkan *error* dari percobaan sebelumnya.

2. Metode K-Fold Cross Validation

Dalam metode ini, data dibagi menjadi K-partisi dan percobaan dilakukan sebanyak K-percobaan. Setiap percobaan menggunakan data partisi ke-K sebagai data *testing* dan sisanya sebagai data *training.* Total *error* ditentukan dengan menjumlahkan semua *erorr* pada setiap percobaan.



Gambar 2.1 K-Fold Cross Validation (Arisona, 2015)

Gambar 2.1 menunjukkan representasi dari metode K-Fold Cross Validation. Setelah membagi data menjadi K-partisi, dilakukan percobaan sebanyak K-percobaan. Pada eksperimen ke-1, data yang digunakan sebagai data *testing* adalah partisi pertama, sedangkan partisi kedua sampai terakhir digunakan sebagai data *training*. Eskperimen dilakukan dengan cara bergantian antara data *testing* dan data *training.*