BAB II

LANDASAN TEORI

2.1 Sistem Pendukung Keputusan

Bonczek, dkk (1980) mendefinisikan sistem pendukung keputusan sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi, yaitu sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (repositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan).

Keputusan yang baik dapat diambil melalui beberapa tahapan fase. Fase yang harus dilalui untuk mengambil suatu keputusan antara lain (Nofriansyah, 2014):

1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendeteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

2. *Design*

Tahap ini merupakan proses menemukan, mengembangkan, dan menganalisis alternative tindakan yang bisa dilakukan. Tahap ini meliputi menguji kelayakan solusi.

3. *Choice*

Tahap ini merupakan tahap proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

Secara garis besar, sistem pendukung keputusan dibangun oleh tiga komponen utama (Nofriansyah, 2014), yaitu:

1. Subsistem Data (*database*)

Subsistem data merupakan komponen sistem pendukung keputusan yang berguna sebagai penyedia data bagi sistem. Data tersebut disimpan untuk diorganisasikan dalam sebuah basis data yang diorganisasikan oleh suatu sistem yang disebut dengan sistem manajemen basis data (database management system).

2. Subsistem model (*modelbase*)

Model adalah suatu tiruan dari alam nyata. Kendala yang sering dihadapi dalam merancang model adalah ketika model yang dirancang tidak mampu mencerminkan seluruh variabel alam nyata, sehingga keputusan yang diambil tidak sesuai dengan kebutuhan. Oleh karena itu, dalam menyimpan model harus memperhatikan dan menjaga fleksibilitasnya.

3. Subsistem dialog (*user system interface*)

Subsistem dialog adalah fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif. Sistem diimplementasikan melalui subsistem dialog sehingga pengguna dapat berkomunikasi dengan sistem yang dibuat.

2.2 Peminatan Prodi Informatika Universitas Multimedia Nusantara

Program studi Informatika Universitas Multimedia Nusantara memiliki beberapa pilihan peminatan yang sesuai dengan visinya untuk menghasilkan lulusan yang berwawasan internasional yang kompeten di bidang ilmu komputer (*computer science*), berjiwa wirausaha dan berbudi pekerti luhur (Universitas Multimedia Nusantara, 2016).

Program peminatan yang tersedia antara lain:

1. Database Lanjutan (Oracle)

Oracle adalah perusahaan multinasional Amerika dalam bidang teknologi computer, terutama spesialisasi dalam membangun dan memasarkan software dan teknologi database (Universitas Multimedia Nusantara, 2016). Pengaruh rangkaian, platform, dan infrastruktur dari aplikasi Oracle, baik teknologi terbaru maupun yang paling dibutuhkan, termasuk kecerdasan buatan (*artificial intelligence*), *machine learning*, *blockchain*, dan *Internet of Things* (IoT), dalam rangka membuat sebuah diferensiasi bisnis dan keuntungan bagi pelanggan (Oracle, 2017).

2. System Applications Products (SAP)

SAP adalah perusahaan software multinasional Jerman dalam bidang operasi manajemen bisnis dan relasi pelanggan. Berdasarkan penguasaan pasar, SAP adalah produsen software independen terbesar ketiga di dunia. Akronim dari SAP adalah *System, Applications, and Products* di bidang pengolahan data. Akronim tersebut berasal dari nama asli dalam bahasa Jerman, yaitu *Systemanalyse und Programmentwicklung* (SAP, 2018).

3. Jaringan Komputer Terapan (Cisco)

Cisco Systems, Inc. adalah perusahaan teknologi multinasional Amerika yang merancang, memproduksi, dan menjual perlengkapan *networking* di seluruh dunia. Perusahaan ini juga menyediakan software dan pelayanan dalam bidang *networking*. Cisco berfokus pada teknologi *networking* berbasis *Internet Protocol* (IP), teknologi *routing* dan *switching* untuk *home networking*, *IP telephony*, *optical networking*, keamanan, *storage area networking*, dan teknologi nirkabel (*wireless*) (Rouse, 2016).

4. Game Development

Game development diterima sebagai salah satu tool yang menarik dalam kurikulum Computer science (Leutenegger dan Edgington, 2007). Merancang dan membangun sebuah game yang dapat dimainkan adalah suatu pekerjaan yang menantang dan paling baik diimplementasikan pada *course* tahap lanjut dimana mahasiswa sudah memiliki pengalaman yang cukup dalam pembuatan piranti lunak dan pengetahuan terhadap topik *computer science* yang lainnya (Kurkovsky, 2013).

5. Keamanan Jaringan (CEH)

Certified Ethical Hacker (CEH) adalah suatu kualifikasi yang didapatkan dengan menilai keamanan dari sistem komputer, menggunakan teknik *penetration testing*. Seseorang yang telah tersertifikasi merupakan seorang professional yang memahami dan mengetahui bagaimana mencari kelemahan dan kerentanan dari suatu sistem target dan menggunakan pengetahuan dan peralatan yang sama seperti seorang peretas (*hacker*) jahat, namun dalam cara yang sah menurut hukum untuk melakukan penilaian keamanan sistem target (EC-Council, 2018).

6. Mobile System Development

Mobile operating system dibuat dan dibangun oleh Apple Inc. dan didistribusikan secara ekslusif untuk perangkat keras (*hardware*) Apple (Universitas Multimedia Nusantara, 2016). Pembangunan aplikasi iPhone saat ini harus dibuat dalam C, C++, atau Objective-C (Wilson, 2010). XCode IDE hanya dapat dijalankan dengan Mac OS X dan bersamaan dengan Simulator iPhone (Apple Inc., 2010).

2.3 Data Mining

Data mining merupakan proses pencarian pola dan relasi-relasi yang tersembunyi dengan tujuan untuk melakukan klasifikasi, estimasi, prediksi, *association rule*, *clustering*, deskripsi dan visualisasi. *Data mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual (Larose 2005). *Data mining*, sering juga disebut sebagai *Knowledge Discovery in Database* (KDD). KDD adalah kegiatan yang meliputi pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola atau hubungan dalam *data set* berukuran besar (Santosa, 2007).

Nama *data mining* sebenarnya mulai dikenal sejak tahun 1990, ketika pekerjaan pemanfaatan data menjadi sesuatu yang penting dalam berbagai bidang, mulai dari bidang akademik, bisnis, hingga medis (Gorunescu, 2011). *Data mining* dapat diterapkan pada berbagai bidang yang mempunyai sejumlah data, tetapi karena wilayah penelitian dengan sejarah yang belum lama, dan belum melewati masa ‘remaja’, maka *data mining* masih diperdebatkan posisi bidang pengetahuan yang memilikinya. Maka, Daryl Pregibon menyatakan bahwa “*data mining* adalah

campuran dari statistik, kecerdasan buatan dan riset basis data" yang masih berkembang (Gorunescu, 2011). Terlepas dari 'remaja'-nya *data mining*, ternyata *data mining* diproyeksikan menjadi jutaan dolar di dunia industri pada tahun 2000, sedangkan pada saat yang sama, ternyata *data mining* dipandang sebelah mata oleh sejumlah peneliti sebagai *dirty word in statistics* (Gorunescu, 2011).

Secara garis besar metode pelatihan yang digunakan dalam teknik-teknik *data mining* dibedakan ke dalam dua pendekatan (Santosa, 2007), yaitu:

1. *Unsupervised learning*, metode ini diterapkan tanpa adanya latihan (*training*) dan tanpa ada guru (*teacher*). Guru disini adalah label dari data.
2. *Supervised learning*, yaitu metode belajar dengan adanya latihan dan pelatih. Beberapa contoh data dalam pendekatan ini yang mempunyai *output* atau label selama proses *training* digunakan untuk menemukan fungsi keputusan, fungsi pemisah atau fungsi regresi.

Ada beberapa teknik yang dimiliki *data mining* berdasarkan tugas yang bisa dilakukan (Larose, 2005), yaitu:

1. Deskripsi

   Para peneliti biasanya mencoba menemukan cara untuk mendeskripsikan pola dan *trend* yang tersembunyi dalam data.

2. Estimasi

   Estimasi mirip dengan klasifikasi,kecuali variabel tujuan yang lebih ke arah numerik dari pada kategori.

3. Prediksi

Prediksi memiliki kemiripan dengan estimasi dan klasifikasi. Hanya saja, prediksi hasilnya menunjukkan sesuatu yang belum terjadi (mungkin terjadi di masa depan).

4. Klasifikasi

Tujuan dalam klasifikasi variabel bersifat kategorik. Misalnya, kita akan mengklasifikasikan pendapatan dalam tiga kelas, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

5. *Clustering*

Clustering lebih ke arah pengelompokan *record*, pengamatan, atau kasus dalam kelas yang memiliki kemiripan.

6. Asosiasi

Mengidentifikasi hubungan antara berbagai peristiwa yang terjadi pada satu waktu.

Secara sistematis, ada tiga langkah utama dalam data mining (Gorunescu, 2011):

1. Eksplorasi atau pemrosesan awal data

Eksplorasi atau pemrosesan awal data terdiri dari permbersihan data, normalisasi data, transformasi data, penanganan data yang salah, reduksi dimensi, pemilihan subset fitur, dan sebagainya.

2. Membangun model dan melakukan validasi terhadapnya

Membangun model dan melakukan validasi terhadapnya berarti melakukan analisis berbagai model dan memilih model dengan kinerja prediksi yang terbaik. Metode-metode yang digunakan dalam langkah ini

seperti klasifikasi, regresi, analisis *cluster*, deteksi anomali, analisis asosiasi, analisis pola sekuensial, dan sebagainya.

3. Penerapan

Penerapan berarti menerapkan model pada data yang baru untuk menghasilkan perkiraan atau prediksi masalah yang diinvestigasi.

2.4 Algoritma Naïve Bayes Classifier

Bayesian classification adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas. *Bayesian classification* didasarkan pada teorema Bayes yang memiliki kemampuan klasifikasi serupa dengan *decision tree* dan *neural network*. *Bayesian classification* terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam database dengan data yang besar (Kusrini dan Luthfi, 2009).

Naïve Bayes Classifier merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Algoritma mengunakan teorema Bayes dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas (Patil dan Sherekar, 2013). Definisi lain mengatakan Naive Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya (Bustami 2014).

Algoritma Naive Bayes menggunakan dasar teorema Bayes (Rumus 2.1) dalam melakukan klasifikasi (Natalius, 2011).

$$P(H|X) = \frac{P(X|H)\ P(H)}{P(X)} \quad \text{... (2.1)}$$

Keterangan:

X = Data dengan class yang belum diketahui

H = Hipotesis data merupakan suatu kelas spesifik

P(H|X) = Probabilitas hipotesis H berdasarkan kondisi X (posteriori probabilitas)

P(H) = Probabilitas hipotesis H (prior probabilitas)

P(X|H) = Probabilitas X berdasarkan kondisi pada hipotesis H

P(X) = Probabilitas X

Persamaan teorema Bayes diatas dapat ditulis secara sederhana menjadi seperti Rumus 2.2 dibawah ini (Natalius, 2011):

$$Posterior = \frac{likelihood\ x\ prior}{evidence} \quad \text{... (2.2)}$$

Algoritma *Naive Bayes Classifier* adalah metode yang didasari oleh probabilitas dan teorema Bayesian. Algoritma ini memiliki asumsi bahwa setiap variabel *x* bersifat bebas. *Posterior* adalah probabilitas masuknya sampel karakteristik tertentu atribut dalam kelas. *Prior* adalah probabilitas munculnya kelas. *Likelihood* adalah probabilitas kemunculan karakteristik-karakteristik atribut sampel terhadap kelas. *Evidence* adalah probabilitas kemunculan karakteristik-karakteristik atribut sampel secara global (Bustami, 2014). *Naive Bayes Classifier* mengansumsikan bahwa keberadaan sebuah atribut tidak ada kaitannya dengan keberadaan atribut yang lain, maka dari itu perhitungan bisa dilakukan seperti pada Rumus 2.3 dibawah ini (Natalius, 2011).

$$P(X|Ci) = \prod_{k=1}^{n} P(Xk|Ci) \quad \text{... (2.3)}$$

Keterangan:

Ci = Kelas-kelas yang ada pada data

X = Data atribut yang kelasnya belum teridentifikasi

P(X|Ci) = Probabilitas atribut terhadap kondisi kelas Ci

P(Xk|Ci) = Probabilitas atribut ke sekian terhadap kondisi kelas Ci

Menurut Raschka (2014) perhitungan prior dapat didapatkan seperti rumus 2.4 dibawah ini.

$$P(Ci) = \frac{N_{Ci}}{N_c} \quad \text{... (2.4)}$$

Keterangan:

Nci = Jumlah seluruh kelas ke-*i*

Nc = Jumlah seluruh data (banyaknya data)

Kelas (label) dari data sampel X adalah kelas (label) yang memiliki nilai maksimum P(X|Ci)*P(Ci), karena P(X|Ci) dapat diketahui melalui persamaan di atas (Abidin 2015). Secara matematis klasifikasi *Naive Bayes Classifier* bisa dirumuskan sebagai berikut (Natalius, 2011):

$$C_{NB} = argmax \prod_{k=1}^{n} P(Xk|Ci)\, P(Ci) \quad \text{... (2.5)}$$

Keterangan:

P(Xk|Ci) = probabilitas Xk terhadap kelas C*i* (posterior)

P(Ci) = probabilitas dari kelas C*i* (prior)

Formula klasifikasi diatas tidak memuat nilai *evidence*, hal ini disebabkan karena evidence memiliki nilai yang positif dan tetap sama untuk semua kelas sehingga tidak mempengaruhi perbandingan nilai *posterior* (Natalius, 2011).

2.5 Laplace Smoothing

Laplace smoothing atau *add-one smoothing* digunakan untuk menghilangkan dugaan parameter yang bernilai nol. Proses perhitungan *naïve bayes classifier* terdapat sedikit keraguan apabila ada peluang yang bernilai nol. Oleh karena itu digunakan *laplace smoothing* yaitu penambahan angka satu

sehingga tidak ada peluang yang akan bernilai nol. Persamaan *laplace smoothing* untuk probabilitas prior dituliskan pada persamaan dibawah ini(Hafilizara, 2014).

$$PLap(y) = \frac{C(y)+1}{N+|y|} \quad \text{... (2.6)}$$

Keterangan:

C(y) = jumlah frekuensi kemunculan kelas ke y

N = jumlah banyaknya data

|y| = jumlah banyaknya jenis kategori

Persamaan *laplace smoothing* untuk probabilitas atribut ke sekian terhadap kelas ke sekian dituliskan pada persamaan dibawah ini(Hafilizara, 2014).

$$PLap(x|y) = \frac{C(x,y)+1}{C(y)+|x|} \quad \text{... (2.7)}$$

Keterangan:

C(x,y) = nilai probabilitas atribut ke x terhadap kelas ke y

C(y) = jumlah frekuensi kemunculan kelas ke y

|x| = jumlah banyaknya jenis karakteristik pada setiap atribut

2.6 Confusion Matrix

Matriks confusion merupakan tabel yang mencatat hasil kerja klasifikasi. Tabel 1 merupakan contoh matriks confusion yang melakukan klasifikasi masalah biner (dua kelas) untuk dua kelas, misalnya kelas 0 dan 1 (Prasetyo 2014).

Tabel 2.1 Matriks Confusion Untuk Klasifikasi Dua Kelas

f_{ij}		Kelas hasil prediksi (j)	
		Kelas = 1	Kelas = 0
Kelas asli (i)	Kelas = 1	TP	FN
	Kelas = 0	FP	TN

Berdasarkan isi matriks confusion, maka dapat diketahui jumlah data dari masing-masing kelas yang diprediksi secara benar yaitu (TP + TN) dan data yang diklasifikasikan secara salah yaitu (FN + FP). Kuantitas matriks confusion dapat diringkas menjadi dua nilai, yaitu akurasi dan laju error. Penggunaan confusion matrix (error matrix) untuk merepresentasikan akurasi telah direkomendasikan oleh banyak peneliti dan seharusnya diterapkan sebagai standar dalam sebuah laporan. *Confusion matrix* adalah susunan angka yang terstruktur persegi yang terdiri dari baris dan kolom yang mewakili jumlah unit sampel. Bagian kolom biasanya berisi data referensi, sedangkan bagian baris mengindikasikan klasifikasi yang dibentuk dari data yang terdaftar (Congalton, 1991). Rumus 2.7 untuk menghitung akurasi dapat dituliskan seperti persamaan dibawah ini (Bramer, 2017).

$$accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad \ldots (2.8)$$

True positives(TP) adalah jumlah *record* positif yang diklasifikasikan sebagai positif, *false positives*(FP) adalah jumlah *record* positif yang diklasifikasikan sebagai negatif, *false negatives*(FN) adalah jumlah *record* negatif yang diklasifikasikan sebagai positif, *true negatives*(TN) adalah jumlah *record* negatif yang diklasifikasikan sebagai negatif (Bramer, 2007).

Kuantitas yang bisa digunakan sebagai metrik kinerja klasifikasi adalah sensitivitas dan *precision*. Kedua kuantitas ini memberikan nilai kinerja yang lebih relevan. Sensitivitas atau *recall* (*true positive rate*) mengukur proporsi positif asli yang dikenali (diprediksi) secara benar. *Precision* mengukur seberapa sering prediksi proporsi positif asli benar (Prasetyo 2014).

Rumus 2.9, 2.10 dan 2.11 adalah rumus untuk mencari nilai *recall* dan *precision* (Bramer, 2007).

$$recall = \frac{TP}{(TP+FN)} \quad \ldots (2.9)$$

$$precision = \frac{TP}{(TP+FP)} \quad \ldots (2.10)$$

Nilai *f-measure* adalah nilai *harmonic* atau nilai rata-rata antara nilai *precision* dan nilai *precision* dan nilai *recall* yang dapat dicari setelah menghitung nilai *recall* dan *precision*. Pencarian nilai *f-measure* menggunakan Rumus 2.11 dibawah ini (Manning *et al*, 2009).

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad \ldots (2.11)$$

		Predicted Number			
		Class 1	Class 2	…	Class *n*
Actual Number	Class 1	x_{11}	x_{12}	…	x_{1n}
	Class 2	x_{21}	x_{22}	…	x_{2n}
	. . .	. . .	. . .	. . .	. . .
	Class *n*	x_{n1}	x_{n2}	…	x_{nn}

Gambar 2.1 Contoh Gambar Confusion Matrix Multi Class (Manliguez, 2016)

Gambar 2.1 menunjukan contoh *confusion matrix* yang digunakan untuk m klasifikasi *multiple class*. Rumus 2.11, 2.12, 2.13 dan 2.14, 2.15 digunakan untuk mencari nilai *true positive*, *true negative*, *false positive*, *false negative* dan *overall accuracy* (Manliguez, 2016).

$$TP_{all} = \sum_{j=1}^{n} x_{jj} \quad \ldots (2.12)$$

Keterangan:

TP_{all} = jumlah seluruh nilai pada diagonal *confusion matrix*

j = indeks baris dan kolom *confusion matrix*

n = total besarnya ukuran baris dan kolom *confusion matrix*

x = isi nilai pada *matrix*

$$TTN_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_j \sum_{\substack{k=1 \\ k \neq i}}^{n} x_k \quad \ldots(2.13)$$

Keterangan:

TTN_i = jumlah *true negative* ke i kecuali baris dan kolom kelas ke i

j = indeks baris *matrix* kecuali indeks baris kelas ke i

k = indeks kolom *matrix* kecuali indeks kolom kelas ke i

x = nilai pada *matrix*

$$TFP_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ji} \quad \ldots (2.14)$$

Keterangan:

TFP_i = jumlah *false positive* ke i kecuali baris kelas ke i dan TP_i

j = indeks baris *matrix* ke i

i = indeks kolom *matrix* kelas ke i

x = nilai pada *matrix*

$$TFN_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} \quad \ldots (2.15)$$

Keterangan:

TFN_i = jumlah *false negative* ke i kecuali kolom kelas ke i dan TP_i

j = indeks baris *matrix* ke i

i = indeks kolom *matrix* kelas ke i

x = nilai pada *matrix*

$$\text{Acc} = \frac{TTP_{all}}{N} \quad \ldots (2.16)$$

Keterangan:

Acc = akurasi *confusion matrix multidimensional*

TTP_{all} = nilai *true positive* seluruh kelas

N = total banyaknya data uji

Nilai metrik kinerja klasifikasi akurasi, *recall*, *precision* dan *f-measure* untuk sistem klasifikasi terbagi menjadi beberapa kelompok seperti berikut (Gorunescu, 2011).

1. 0,90-1,00 = klasifikasi sangat baik
2. 0,80-0,90 = klasifikasi baik
3. 0,70-0,80 = klasifikasi cukup
4. 0,60-0,70 = klasifikasi buruk
5. 0,50-0,60 = klasifikasi salah