MODUL DATA MINING
KLASIFIKASI
PERTEMUAN 8 (ONLINE)

Disusun Oleh
Syefira Salsabila

PENDAHULUAN

Berkembangnya **Ilmu Data Mining** memberikan inovasi baru dalam hal pendayagunaan kumpulan data yang banyak sehingga dapat bermanfaat bagi pengembangan pengetahuan, baik secara khusus pada bidang yang berkaitan dengan data tersebut maupun secara global. Banyak fungsi yang dapat diterapkan dari ilmu data mining antara lain, estimasi, prediksi, klasterisasi, klasifikasi dan asosiasi. Untuk mencapai fungsi-fungsi tersebut dilakukan dengan berbagai metode (algoritma) seperti regresi untuk estimasi, *Support Vector Macsine* (SVM) untuk prediksi, *KMeans* untuk klasterisasi, C4.5 untuk klasifikasi, *apriori* untuk asosiasi.

Data mining atau lebih di kenal juga dengan sebutan knowledge discovery in databases (KDD). Data mining merupakan salah satu cara yang digunakan untuk mendapatkan pengetahuan baru dengan memanfaatkan jumlah data yang sangat besar. Kegiatan dalam Data mining meliputi pengumpulan dan pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data yang berukuran besar. Beberapa teknik telah dikembangkan dan diimplementasikan untuk mengekstrak pengetahuan dan informasi untuk menemukan pola pengetahuan yang mungkin berguna untuk pengambilan keputusan. Teknik-teknik yang digunakan untuk pengekstrakan pengetahuan dalam data mining adalah pengenalan pola, clustering, asosiasi, prediksi dan klasifikasi.

Salah satu pendekatan yang dapat digunakan untuk menganalisis sekumpulan data adalah klasifikasi. Klasifikasi merupakan salah satu teknik data mining yang digunakan untuk membangun suatu model dari sampel data yang belum terklasifikasi untuk digunakan mengklasifikasi sampel data baru ke dalam kelas-kelas yang sejenis. Klasifikasi adalah pemprosesan untuk menemukan sebuah model atau fungsi yang menjelaskan dan mencirikan konsep atau kelas data, untuk kepentingan tertentu. Keluaran yang dihasilkan oleh klasifikasi *data mining* dapat digunakan untuk memperbaiki pengambilan keputusan bagi analis.

Klasifikasi termasuk ke dalam *supervised learning* karena menggunakan sekumpulan data untuk dianalisis terlebih dahulu, kemudian pola dari hasil analisis tersebut digunakan untuk pengklasifikasian data uji. Proses klasifikasi data terdiri dari pembelajaran dan klasifikasi. Pada pembelajaran data *training* dianalisis menggunakan algoritma klasifikasi, selanjutnya pada klasifikasi digunakan data *testing* untuk memastikan tingkat akurasi dari *rule* klasifikasi yang digunakan. Teknik klasifikasi dibagi menjadi lima kategori berdasarkan perbedaan konsep matematika, yaitu berbasis statistik, berbasis jarak, berbasis pohon keputusan, berbasis jaringan syaraf, dan berbasis *rule*. Ada banyak algoritma dari masing-masing kategori tersebut, namun yang populer dan sering digunakan diantaranya yaitu *naive bayes*, *nearest neighbour* dan *decision tree.*

Salah satu penerapan ilmu data mining, yaitu pada permasalahan penumpukan data rekam medis di Rumah Sakit. Rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang diberikan kepada pasien. Rekam medis harus dibuat secara

tertulis, lengkap, dan jelas atau secara elektronik. Penyelenggaraan rekam medis dengan menggunakan teknologi informasi elektronik diatur oleh peraturan tersendiri. Informasi dalam rekam medis dijaga kerahasiaannya oleh dokter, tenaga kesehatan dan petugas pengelola serta pimpinan sarana pelayanan kesehatan. Data rekam medis terus terakumulasi setiap hari seiring dengan aktivitas rumah sakit. Pemanfaatan rekam medis dapat dipakai sebagai: (1) pemeliharaan kesehatan dan pengobatan pasien; (2) alat bukti dalam proses penegakkan hukum, disiplin kedokteran dan kedokteran gigi, dan penegakkan etika kedokteran dan kedokteran gigi; (3) keperluan pendidikan dan penelitian; (4) dasar pembayaran biaya pelayanan kesehatan; (5) data statistik kesehatan.

Pada dasarnya data mining mempunyai kegunaan serta tugas untuk mengspesifikasikan pola yang harus ditemukan dalam proses data mining. Secara umum tugas data mining dapat dibagi menjadi dua kategori yaitu:

a. Prediktif

Tujuan dari tugas prediktif adalah untuk memprediksi nilai dari atribut tertentu berdasarkan pada nilai dari atribut-atribut lainnya. Atribut yang diprediksi umumnya dikenal sebagai target atau variable tak bebas, sedangkan atribut-atribut yang digunakan untuk membuat prediksi dikenal sebagai variabel bebas.

b. Deskriptif

Tujuan dari tugas deskriptif adalah menurunkan pola-pola (korelasi. Trend, cluster, trayektori, dan anomali) yang meringkas hubungan yang pokok dalam data. Tugas data mining deskriptif sering disebut sebagai penyelidikan dan seringkali memerlukan teknik *postprocessing* untuk validasi dan penjelasan hasil.

Data mining model dibuat berdasarkan slah satu dari dua jenis pembelajaran *supervised* dan *unsupervised.* Fungsi pembelajaran Supervised digunakan untuk memprediksi suatu nilai. Fungsi pembelajarn unsupervised digunakan untuk mencari struktur intrinsik, relasi dalam suatu data yang tidak memerlukan class atau label sebelum dilakukan proses pembelajaran. Contoh dari algoritma pembelajaran unsupervised, diantaranya k-means clustering dan Apriori association rules. Contoh dari algoritma pembeljaran supervised yaitu NaïveBayes untuk klasifikasi.

Metode data mining dapat diklasifikasikan berdasarkan fungsi yang dilakukan atau berdasarkan jenis aplikasi yang menggunakannya:

a. Klasifikasi (supervised)
b. Clustering (unsupervised)
c. Association rules (unsupervised)
d. Attribute importance (supervised)

Klasifikasi

Klasifikasi (Supervised)

Pada persoalan klasifikasi, kita memiliki sejumlah kasus (sampel data) dan ingin memprediksi beberapa class yang ada pada sampel data tersebut. Tiap instan data berisi banyak atribut, dimana masing-masing atribut satu dari beberapa kemungkinan nilai. Hanya satu atribut diantara banyak atribut tersebut yang disebut dengan atribut target, sedangkan atribut yang lain disebut sebagai atribut predictor. Tiap kemungkinan nilai yang dimiliki oleh atribut target menunjukkan class yang diprediksi berdasarkan nilai-nilai dari atribut predictor. Metode klasifikasi digunakan untuk membantu dalam memahami pengelompokan data.

Klasifikasi merupakan salah satu metode dari *data mining.* Klasifikasi adalah metode prediktif yang melakukan pembelajaran terhadap data-data yang sudah ada sehingga menghasilkan suatu model yang digunakan untuk memprediksi data-data baru. Klasifikasi data terdiri dari dua proses yaitu tahap pembelajaran dan tahap pengklasifikasian. Tahap pembelajaran merupakan tahapan dalam pembentukan model klasifikasi, sedangkan tahap pengklasifikasian merupakan tahapan penggunaan model klasifikasi untuk memprediksi label kelas dari suatu data. Contoh sederhana dari teknik *data mining* klasifikasi adalah pengklasifikasian hewan berdasarkan atribut jumlah kaki, habitat dan organ pernafasannya akan diklasifikasikan ke dalam dua label kelas yaitu unggas dan ikan. Label kelas unggas adalah data yang memiliki jumlah kaki dua, habitatnya di darat, dan organ pernafasannya menggunakan paru-paru, sedangkan label kelas ikan adalah data yang memiliki jumlah kaki nol (tidak memiliki kaki), habitat di air, dan organ pernafasannya menggunakan insang. Banyak algoritme yang dapat digunakan dalam pengklasifikasian data, namun dalam penelitian ini hanya akan membandingkan tiga algoritma saja, yakni *naive bayes*, *nearest neighbour*, dan *decision tree.*

A. *K-Nearest Neighbor* (KNN)

Nearest Neighbour adalah algoritma pengklasifikasian yang didasarkan pada analogi, yaitu membandingkan data uji dengan data pelatihan yang berada dekat dengan dan memiliki kemiripan dengan data uji tersebut. KNN merupakan metode yang menggunakan algoritma *supervised* dimana hasil dari *query instance* yang baru diklasifikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru bedasarkan atribut dan *training sample*. Algoritma KNN sangatlah sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan KNN-nya dan mudah untuk di implementasikan. KNN memiliki kemampuan kerja yang rendah ketika *training dataset* besar. Salah satu masalah pada algoritma ini adalah bobot yang sama dari semua atribut dalam menghitung jarak antara data *testing* dan data *training*, bagaimana pun, mungkin dari semua atribut ada beberapa atribut yang kurang penting untuk proses klasifikasi dan ada beberapa atribut yang lebih penting untuk proses klasifikasi. Sehingga tidak jelas jarak mana yang harus digunakan dan atribut mana yang harus

digunakan untuk mendapatkan hasil terbaik. Hal ini dapat menyesatkan proses klasifikasi dan dapat menurunkan akurasi dari klasifikasi. Pendekatan yang banyak dilakukan untuk mengatasi masalah ini adalah dengan memberi bobot yang berbeda pada tiap-tiap atribut ketika mengukur jarak dua record. Pembobotan berguna untuk menentukan jarak antar atribut tetangga dengan record baru berdasarkan similarity.

Ketepatan algoritma k-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevan isinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik. KNN juga merupakan contoh teknik lazy learning, yaitu teknik yang menunggu sampai pertanyaan (query) datang agar sama dengan data training.

Kemiripan data uji dengan data pelatihan didasarkan pada jaraknya. Banyak persamaan yang dapat digunakan untuk menghitung jarak antara data uji dan data pelatihan. Tiga diantaranya yang paling sering digunakan adalah:

a. Atribut yang bertipe numerik

Terdapat dua pendekatan perhitungan jarak/kemiripan yang umum digunakan untuk atribut yang bertipe numerik, yaitu *euclidean distance* dengan persamaan berikut:

$$Dist(x1, x2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})} \quad (2)$$

Keterangan:
n : jumlah data
x1 : data uji
x2 : data pembelajaran

Persamaan yang kedua yaitu *Manhattan distance* sebagai berikut :

$$Dist(p_i(am), p_i(nc)) = \frac{p_i(am) - p_i(nc)}{\max\_dist\_i} \quad (3)$$

Keterangan:
pi : atribut ke-i
an : data pembelajaran
nc : data uji

b. Atribut yang bertipe simbolik

Persamaan yang digunakan untuk atribut yang menggunakan istilah eksplisit yaitu ada atau tidak ada, memiliki atau tidak memiliki, ya atau tidak dan sebagainya maka perhitungan kemiripan atau jarak dapat dihitung dengan fungsi sebagai berikut:

$$Sim(Ki(a), Ki(b)) = \begin{cases} 0 & Ki(a) \neq Ki(b) \\ 1 & Ki(a) = Ki(b) \end{cases} \quad (4)$$

Keterangan :
Ki(a) : kriteria ke-i dari kasus a
Ki(b) : kriteria ke-i dari kasus b
Sim(Ki(a),Ki(b) : nilai kemiripan kriteria ke-i antara kasus a dengan kasus b

Perhitungan selanjutnya adalah persamaan untuk mencari kemiripan dengan *nearest neighbor* yaitu:

$$Similarity(T, S) = \frac{\sum_{i=1}^{n} Sim(K_i(T), K_i(S)) x w_i}{\sum_{i=i}^{n} w} \quad (5)$$

Keterangan:
T : data uji
S : data pembelajaran
n : jumlah kriteria
w : bobot kriteria
Sim(*K* (*T*), *K* (*S*)) *i i* : Nilai kemiripan/jarak kriteria kasus target dan target sumber

B. Naive Bayes

Naive Bayes merupakan pengklasifikasian dengan metode probabilitas yang ditemukan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai teorema Bayes. Teorema tersebut dikombinasikan dengan *naive* di mana diasumsikan kondisi antar petunjuk (atribut) saling bebas. Klasifikasi *naive* Bayes diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya. Salah satu pengaplikasian dari *naive* Bayes yaitu pada bidang kesehatan.

Teorema bayes adalah perhitungan statistik dengan menghitung probabilitas kemiripan kasus lama yang ada dibasis kasus dengan kasus baru. *Teorema bayes* memiliki tingkat akurasi yang tinggi dan kecepatan yang baik ketika diterapkan pada *database* yang besar. *Naïve bayes* termasuk ke dalam pembelajaran *supervised*, sehingga pada tahapan pembelajaran dibutuhkan data awal berupa data pelatihan untuk dapat mengambil keputusan. Pada tahapan pengklasifikasian akan dihitung nilai probabilitas dari masing-masing label kelas yang ada terhadap masukan yang diberikan. Label kelas yang memiliki nilai probabilitas paling besar yang akan dijadikan label kelas data masukan tersebut. *Naive bayes* merupakan perhitungan *teorema bayes* yang paling sederhana, karena mampu mengurangi kompleksitas komputasi menjadi multiplikasi sederhana dari probabilitas. Selain itu, algoritma *naive bayes* juga mampu menangani set data yang memiliki banyak atribut. Persamaan dari *naive bayes* sebagai berikut:

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)} \quad (1)$$

Keterangan :
X : Kriteria suatu kasus berdasarkan masukan
Ci : Kelas solusi pola ke-i, dimana i adalah jumlah label kelas
P(Ci|X) : Probabilitas kemunculan label kelas Ci dengan kriteria masukan X
P(X|Ci) : Probabilitas kriteria masukan X dengan label kelas Ci
P(Ci) : Probabilitas label kelas Ci

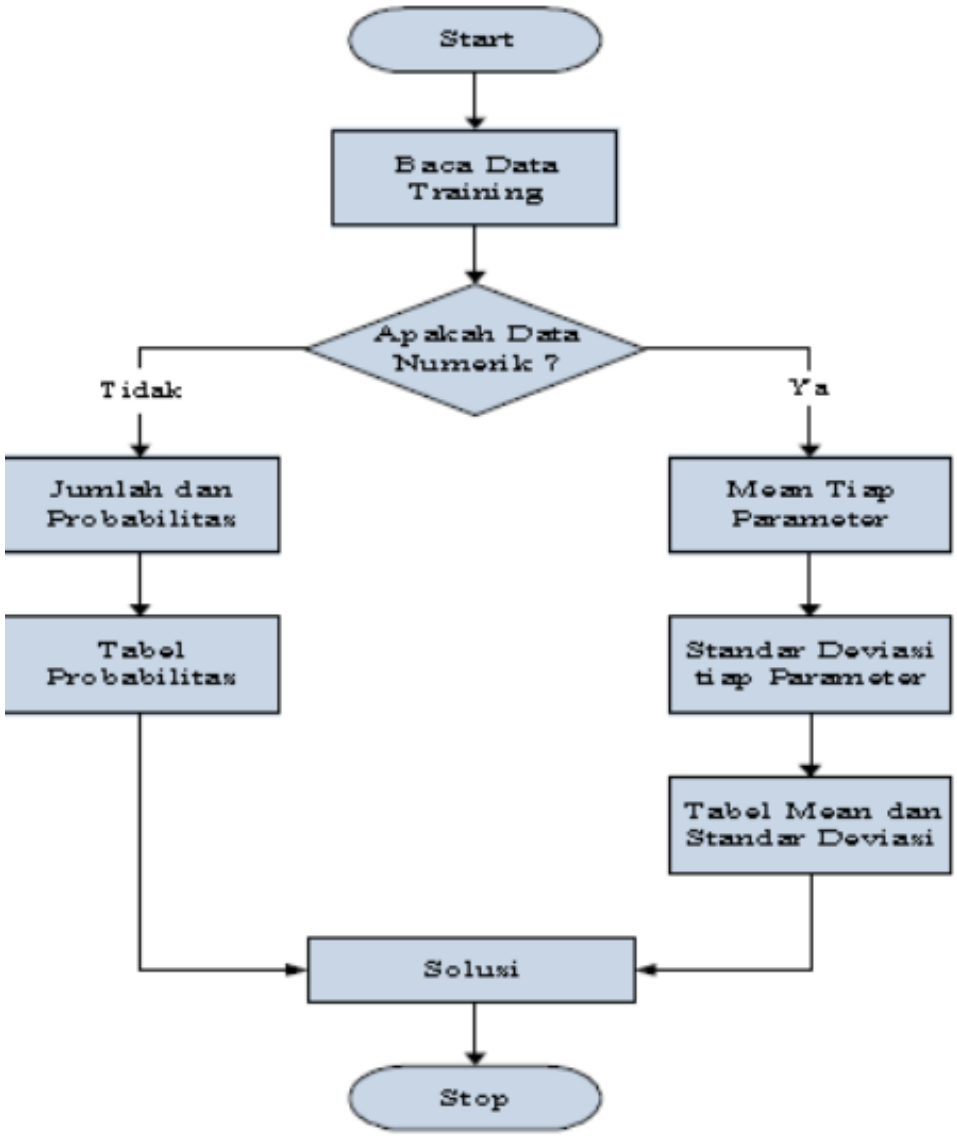
Naive Bayes mendasarkan pada asumsi penyederhanaan dimana nilai atribut secara kondisional saling bebas apabila diberikan nilai *output.* Metode ini merupakan sebuah metode yang berakar pada teorema *Bayes*. Persamaan (2) merupakan persamaan Teorema *Bayes* yang menyatakan bahwa:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (2)$$

Keterangan :

P(B|A)	=	probabilitas posterior, probabilitas muncul B jika diketahui A
P(B|A)	=	probabilitas posterior, probabilitas muncul A jika diketahui B
P(A)	=	probabilitas prior, probabilitas kejadian A
P(B)	=	probabilitas prior, probabilitas kejadian B

Model Naïve Bayes adalah klasifikasi statistik yang dapat digunakan untuk memprediksi suatu kelas. Model Naïve Bayes dapat diasumsikan bahwa efek dari suatu nilai atribut sebuah kelas yang diberikan adalah bebas dari atribut-atribut lain. Naive Bayes memiliki alur seperti pada gambar dibawah ini.



Gambar. Alur Naive Bayes

Kelebihan yang dimiliki oleh Naïve Bayes adalah dapat menangani data kuantitatif dan data diskrit, Naïve Bayes kokoh terhadap noise, Naïve Bayes hanya memerlukan sejumlah kecil data pelatihan untuk mengestimasi parameter yang dibutuhkan untuk klasifikasi, Naïve Bayes dapat menangani nilai yang hilang dengan mengabaikan instansi selama perhitungan estimasi peluang, Naïve Bayes cepat dan efisiensi ruang.

Metode ini penting karena beberapa alasan, termasuk berikut. Hal ini sangat mudah untuk membangun, tidak perlu ada yang rumit Parameter estimasi skema berulang. Ini berarti dapat segera diterapkan untuk besar Data set. Sangat mudah untuk menafsirkan, sehingga pengguna tidak terampil dalam teknologi classifier dapat memahami mengapa itu adalah membuat klasifikasi itu membuat. Dan, sangat penting, hal itu sering sangat baik: Ini mungkin bukan classifier terbaik dalam setiap diberikan aplikasi, tetapi biasanya dapat diandalkan untuk menjadi kuat dan melakukan dengan sangat baik

C. Decision Tree

Algoritma *decision tree* merupakan algoritma yang umum digunakan untuk pengambilan keputusan. *Decision tree* akan mencari solusi permasalahan dengan menjadikan kriteria sebagai *node* yang saling berhubungan membentuk seperti struktur pohon. *Decision tree* adalah model prediksi terhadap suatu keputusan menggunakan struktur hirarki atau pohon. Setiap pohon memiliki cabang, cabang mewakili suatu atribut yang harus dipenuhi untuk menuju cabang selanjutnya hingga berakhir di daun (tidak ada cabang lagi). Konsep data dalam *decision tree* adalah data dinyatakan dalam bentuk tabel yang terdiri dari atribut dan *record.* Atribut digunakan sebagai parameter yang dibuat sebagai kriteria dalam pembuatan pohon.

Proses dalam *decision tree* adalah sebagai berikut:

a. Mengubah bentuk data (tabel) menjadi model pohon

Hal yang dilakukan pada tahapan ini adalah menentukan atribut yang terpilih mulai dari akar, cabang hingga menuju keputusan. Banyak pendekatan yang dapat digunakan untuk menentukan atribut terpilih, pada penelitian ini akan menggunakan perhitungan *gainratio* dari setiap kriteria dengan data sampel. Untuk menghitung nilai *gainratio* dapat dilakukan dengan persamaan sebagai berikut:

$$Gainratio(S,A) = \frac{Gain(S,A)}{SplitInformation(S,A)} \quad (6)$$

Dimana nilai *information gain* bermakna seberapa banyak informasi yang diperoleh dengan mengetahui nilai suatu atribut sedangkan nilai *split information* digunakan untuk suatu atribut yang memiliki banyak *instance* (lebih dari dua dan beragam)

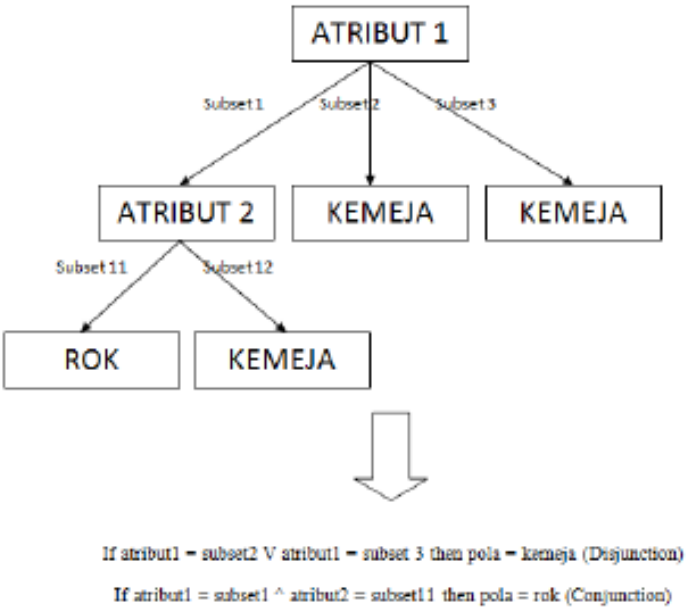
b. Mengubah model pohon menjadi *rule*

Formula untuk membangkitkan *rule* didefinisikan sebagai berikut:

IF premis THEN konklusi

Simpul akar dan cabang akan menjadi premis dari aturan, sedangkan simpul daun akan menjadi bagian dari konklusinya (solusi). Tiap premis yang

terdapat dalam satu atribut akan dihubungkan dengan hubungan disjungsi, sedangkan premis yang memiliki lanjutan premis pada cabang selanjutnya akan dihubungkan dengan konjungsi.



Gambar. Proses Model Pohon Menjadi Rule

c. Menyederhanakan *rule* (*Pruning*)
Pada proses penyederhanaan *rule*, tahapan-tahapan dilakukan sebagai berikut:
a) Membuat tabel distibusi terpadu dengan menyatakan semua nilai kejadian pada setiap *rule*.
b) Menghitung tingkat *independensi* antara kriteria pada suatu *rule*, yaitu antara atribut dengan target atribut (perhitungan tingkat *independensi* menggunakan *test of independency Chi- Square*).
c) Mengeliminasi kriteria yang dianggap tidak perlu, yaitu yang memiliki tingkat *independensi* tinggi.

Misalkan yang ingin dilihat adalah pengaruh jenis pakaian terhadap penentuan solusi pola pakaian yang dapat dibuat, tentukan terlebih dahulu tingkat signifikansinya, sehingga dapat dihitung *degree of freedom* dengan persamaan berikut :

$$\{0.05;(r-1)*(c-1)\} \quad (8)$$

Keterangan:

r : jumlah baris
c : jumlah kolom
Setelah diperoleh nilainya maka dapat dilihat pada tabel untuk memperoleh nilai X^2_{tabel} untuk dibandingkan dengan X^2_{hitung}. X^2_{hitung} diperoleh melalui persamaan berikut :

$$X^2_{hitung} = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{ij} - e_{ij})^2}{e_{ij}} \quad (9)$$

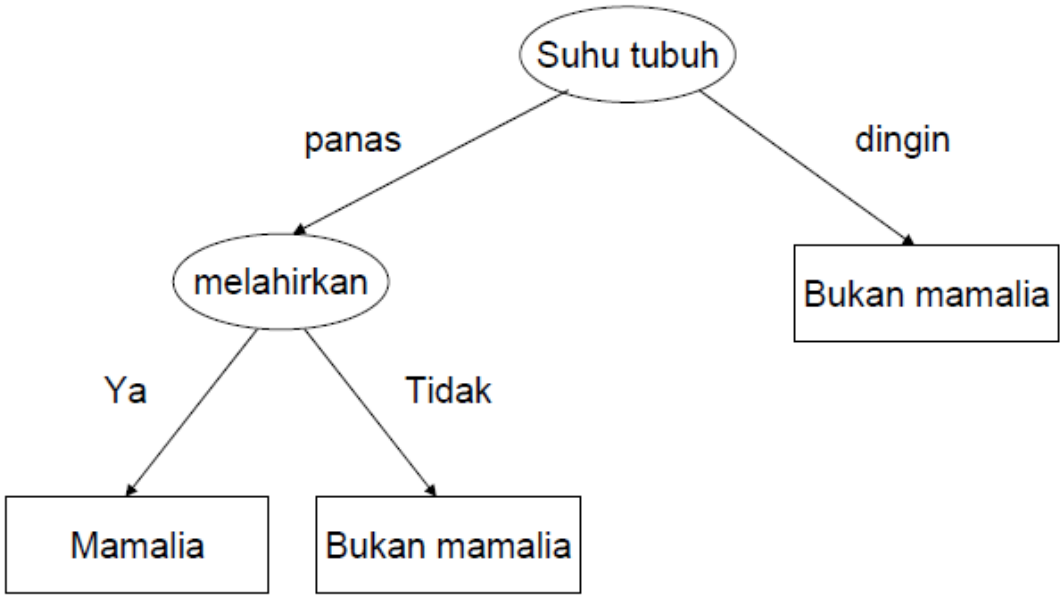
Keterangan:
n_{ij} : nilai *record* baris ke i kolom ke j dari tabel distribusi terpadu.
Sedangkan nilai e_{ij} diperoleh melalui persamaan berikut:

$$e_{ij} = \frac{n_i \bullet n_j}{n} \quad (10)$$

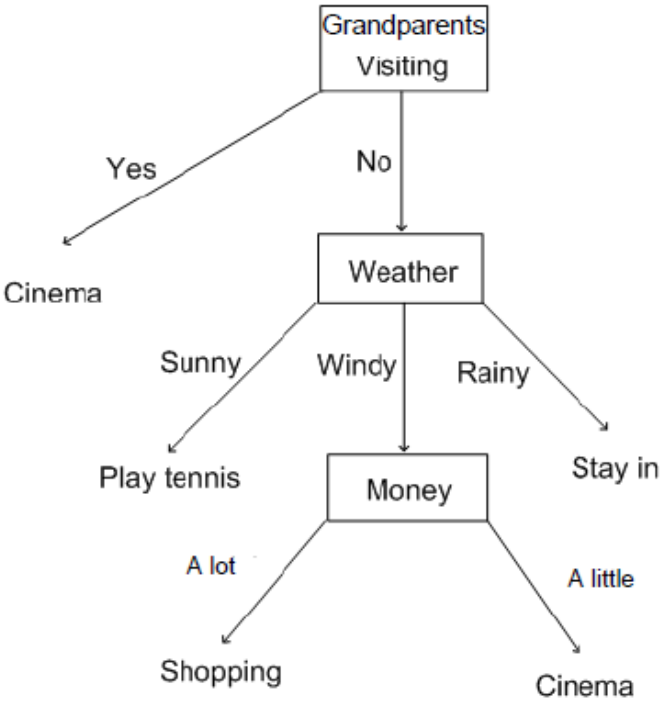
Keterangan:
n_i : marjinal dari baris ke i
n_j : marjinal dari kolom ke j
n : jumlah *record* data

Jika nilai $X^2_{hitung} \leq X^2_{tabel}$ artinya atribut tersebut tidak mempengaruhi atribut target, sehingga *rule* dari atribut tersebut dapat dihilangkan. Namun sebaliknya jika nilai $X^2_{hitung} > X^2_{tabel}$ berarti atribut tersebut mempengaruhi atribut target, sehingga *rule* dari atribut tersebut tidak dapat dihilangkan.

Decision tree adalah bentuk sederhana teknik klasifikasi pada sekumpulan kelas tak berhingga yang direpresentasikan ke dalam bentuk simpul *(node)* dan rusuk *(edge)*. Biasanya, *Decision Tree* dipilih untuk menyelesaikan masalah dengan *output* yang bernilai diskrit.

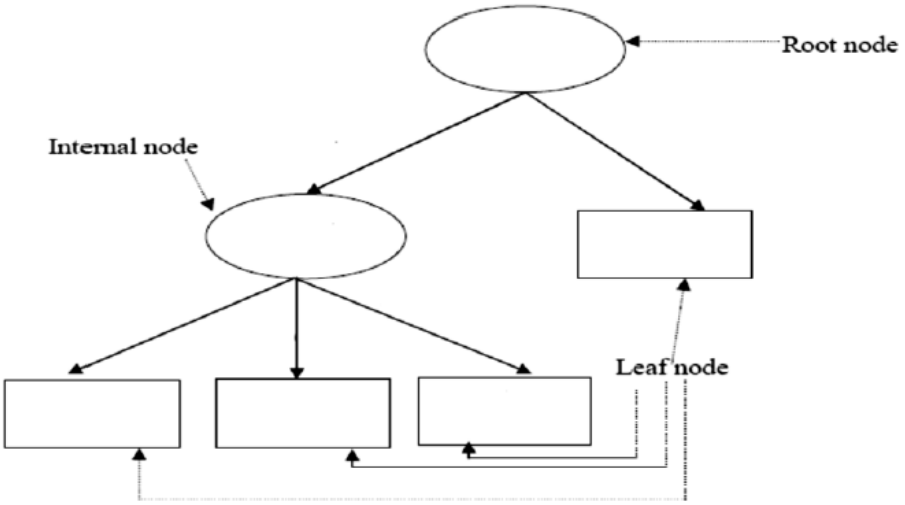

Suhu tubuh
panas
dingin
melahirkan
Bukan mamalia
Ya
Tidak
Mamalia
Bukan mamalia

Contoh



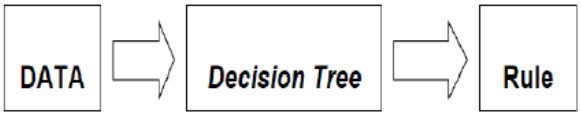
Klasifikasi

Decision tree merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode *decision tree* mengubah fakta yang sangat besar menjadi pohon keputusan yang mempresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami. Dan mereka juga dapat diekspresikan dalam bentuk basis data seperti *Structure Query Language* (SQL) untuk mencari *record* pada data tertentu. Sebuah *decision tree* adalah sebuah struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan-himpunan *record* yang lebih kecil dengan menerapkan serangkaian aturan keputusan. Pada *decision tree* setiap simpul daun menandai label kelas. Simpul yang bukan simpul akhir terdiri dari akar dan simpul internal yang terdiri dari kondisi tes atribut pada sebagian *record* yang mempunyai karakteristik yang berbeda. Simpul akar dan simpul internal ditandai dengan bentuk oval dan simpul daun ditandai dengan bentuk segi empat.



Gambar 4. Struktur *decison tree*

Berikut adalah konsep *decision tree* seperti yang ditunjukkan pada Gambar 5.



Gambar 5. Konsep *decision tree*

Ada beberapa konsep dalam *decision tree*, antara lain:

a. Data dinyatakan dalam bentuk Tabel dengan *atribut* dan *record*.
b. *Atribut* menyatakan suatu parameter yang dibuat sebagai kriteria dalam pembentukan *tree*. Misalkan untuk menentukan main tenis, kriteria yang diperhatikan adalah cuaca, angin dan temperatur. Salah satu atribut merupakan atribut yang menyatakan data solusi *per-item* data yang disebut dengan target atribut.
c. Atribut memiliki nilai-nilai yang dinamakan dengan *instance*.

DAFTAR PUSTAKA

Defiyanti, S., & Jajuli, M. (2015). Integrasi Metode Klasifikasi Dan Clustering dalam Data Mining. *Konferensi Nasional Informatika (KNIF)*, 39-44.

Dewi, S. (2016). KOMPARASI 5 METODE ALGORITMA KLASIFIKASI DATA MINING PADA PREDIKSI KEBERHASILAN PEMASARAN PRODUK LAYANAN PERBANKAN. *Jurnal Techno Nusa Mandiri*, *13*(1), 60-65.

Irwansyah, Edy. Advance Clustering : Teori Dan Aplikasi. jakarta : Bina Nusantara University, 2015. 978-602-280-500-7.

Penerapan Algoritma Naive Bayes untuk Mengklasifikasikan Data Nasabah Asuransi. Bustami.Aceh : TECHSE Jurnal Penelitian Teknik Informatika. 5.

Sabransyah, M., Nasution, Y. N., & Amijaya, F. D. T. (2017). Aplikasi Metode Naive Bayes dalam Prediksi Risiko Penyakit Jantung. *JURNAL EKSPONENSIAL*, *8*(2), 111-118.

Shukla, A.Tiwari, R., & Kala, R. 2010. RealLife Application of Soft Computing. Taylor and Francis Groups, LLC.