BAB 2

TINJAUAN PUSTAKA

2.1. Algoritma Pohon Keputusan

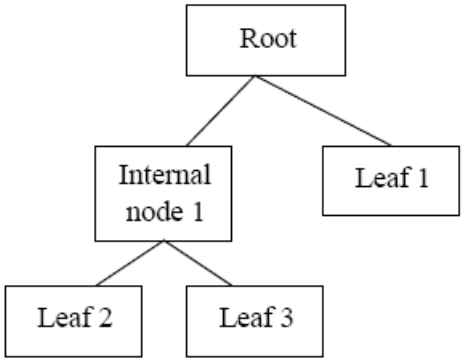
Salah satu metode *data mining* yang umum digunakan adalah pohon keputusan. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan *rule*. Pohon keputusan adalah salah satu metode klasifikasi yang paling populer karena mudah untuk diinterpretasikan oleh manusia. Konsep dari pohon keputusan adalah mengubah data menjadi model pohon keputusan (*decision tree*) dan aturan-aturan (*rule*), ditunjukkan pada Gambar 2.1 berikut.

Gambar 2.1. Konsep Pohon Keputusan

Data dalam pohon keputusan biasanya dinyatakan dalam bentuk tabel dengan atribut dan *record*. Atribut menyatakan suatu parameter yang dibuat sebagai kriteria dalam pembentukan *tree*. Misalkan, untuk menentukan bermain tenis atau tidak, 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. Atribut memiliki nilai-nilai yang dinamakan dengan *instance*. Misalkan atribut cuaca mempunyai *instances* berupa cerah, berawan, dan hujan.

Proses pada pohon keputusan adalah mengubah bentuk data (tabel) menjadi model pohon keputusan, mengubah model pohon keputusan menjadi aturan (Jayanti, et al., 2008). Manfaat utama dari penggunaan pohon keputusan adalah kemampuannya untuk mem-*break down* proses pengambilan keputusan yang kompleks menjadi lebih sederhana sehingga pengambilan keputusan akan lebih menginterpretasikan solusi permasalahan. Pohon keputusan juga berguna untuk mengeksplorasi data, yaitu menemukan hubungan tersembunyi antara sejumlah calon variabel *input* dengan sebuah variabel target.

Pohon keputusan merupakan himpunan aturan IF...THEN. Setiap *path* dalam *tree* dihubungkan dengan sebuah aturan, dimana premis terdiri atas sekumpulan *node-node* yang ditemui, dan kesimpulan dari aturan terdiri atas kelas yang terhubung dengan *leaf* dari *path*. Gambar 2.2 menunjukkan struktur pohon keputusan.



Gambar 2.2. Struktur Pohon Keputusan (Defiyanti & Pardede, 2010)

Bagian awal dari pohon keputusan ini adalah titik akar (*root*), sedangkan setiap cabang dari pohon keputusan merupakan pembagian berdasarkan hasil uji dan titik akhir (*leaf*) merupakan pembagian kelas yang dihasilkan.

Pohon keputusan mempunyai tiga (3) tipe simpul, yaitu:

1. Simpul akar (*root*) → tidak memiliki cabang yang masuk dan memiliki cabang lebih dari satu, terkadang tidak memiliki cabang sama sekali. Simpul ini biasanya berupa atribut yang paling memiliki pengaruh terbesar pada suatu kelas tertentu.
2. Simpul internal (*node*) → hanya memiliki satu cabang yang masuk dan memiliki lebih dari satu cabang yang keluar.
3. Simpul daun (*leaf*) → simpul akhir yang hanya memiliki satu cabang yang masuk dan tidak memiliki cabang sama sekali sekaligus menandai bahwa simpul tersebut merupakan label kelas.

Tahapan awal adalah pengujian simpul akar. Jika pada pengujian simpul akar menghasilkan sesuatu, maka proses pengujian juga dilakukan pada setiap cabang berdasarkan hasil dari pengujian. Hal ini berlaku juga untuk simpul internal dimana suatu kondisi pengujian baru akan diterapkan pada simpul daun. Pada umumnya proses dari sistem pohon keputusan adalah mengadopsi strategi pencarian *top-down* untuk solusi ruang pencariannya. Pada proses mengklasifikasikan sampel yang tidak

diketahui, nilai atribut akan diuji pada pohon keputusan dengan cara melacak jalur dari titik akar sampai titik akhir, kemudian akan diprediksikan kelas yang ditempati sampel baru tersebut.

Pohon keputusan banyak digunakan dalam proses *data mining* karena memiliki beberapa kelebihan, yaitu:

1. Tidak memerlukan biaya yang mahal saat membangun algoritma.
2. Mudah untuk diinterpetasikan.
3. Mengakomodasi data yang hilang.
4. Mudah mengintegrasikan dengan sistem basis data.
5. Memiliki nilai ketelitian yang lebih baik.
6. Dapat menemukan hubungan tak terduga dan suatu data.
7. Dapat menggunakan data pasti/mutlak atau data kontinu.

2.1.1. Pohon Keputusan ID3

Algoritma ID3 atau *Iterative Dichotomiser 3 (ID3)* merupakan sebuah metode yang digunakan untuk membuat pohon keputusan yang telah dikembangkan oleh J. Ross Quinlan pada sekitar akhir 1970-an dan awal 1980-an (Han, et al., 2006). Algoritma pada metode ini menggunakan konsep dari *entropy* informasi. Algoritma ini melakukan pencarian secara rakus/menyeluruh (*greedy*) pada semua kemungkinan pohon keputusan.

Secara ringkas, langkah kerja algoritma ID3 dapat digambarkan sebagai berikut (Defiyanti & Pardede, 2010):

1. Hitung *entropy* dan *information gain* dari setiap atribut dengan menggunakan rumus:

$$Entropy\ (S) = -P_{+}log_2 P_{+} - P_{-}log_2 P_{-} \quad \ldots (2.1)$$

dimana:

S = ruang (data) sampel yang digunakan untuk *training*

P_+ = jumlah yang bersolusi positif (mendukung) pada data sampel untuk kriteria tertentu

P_- = jumlah yang bersolusi negatif (tidak mendukung) pada data sampel untuk kriteria tertentu

$$Gain(S, A) = Entropy(S) - \sum_{V \in nilai(A)} \frac{|S_v|}{|S|} Entropy\ (S_v) \qquad \ldots (2.2)$$

dimana:

S = ruang (data) sampel yang digunakan untuk *training*

A = atribut

V = suatu nilai yang mungkin untuk atribut A

Nilai(A) = himpunan yang mungkin untuk atribut A

$|S_v|$ = jumlah sampel untuk nilai V

|S| = jumlah seluruh sampel data

Entropy(S_v) = *entropy* untuk sampel-sampel yang memiliki nilai V

Tujuan dari pengukuran nilai *information gain* adalah untuk memilih atribut yang akan dijadikan cabang pada pembentukan pohon keputusan. Dalam hal ini, dipilih atribut yang memiliki nilai *information gain* terbesar.

2. Bentuk simpul yang berisi atribut tersebut.
3. Ulangi proses perhitungan *information gain* yang akan terus dilaksanakan sampai semua data telah termasuk dalam kelas yang sama. Atribut yang telah dipilih tidak diikutkan lagi dalam perhitungan nilai *information gain*.

ID3 berhenti jika atribut sempurna mengklasifikasikan *training sets* atau secara rekursif mengoperasikan nilai n, dimana n adalah banyaknya nilai kemungkinan dari suatu untuk mendapatkan atribut terbaik.

Adapun sampel data yang digunakan oleh ID3 memiliki beberapa syarat, yaitu:

a) Deskripsi atribut-nilai. Atribut yang sama harus mendeskripsikan tiap contoh dan memiliki jumlah nilai yang sudah ditentukan.
b) Kelas yang sudah didefinisikan sebelumnya. Suatu atribut contoh harus sudah didefinisikan, karena mereka tidak dipelajari oleh ID3.
c) Kelas-kelas yang diskrit. Kelas harus digambarkan dengan jelas. Kelas yang kontinu dipecah-pecah menjadi kategori-kategori yang relatif, misalnya saja metal dikategorikan menjadi "*hard, quite hard, flexible, soft, quite soft*".

d) Jumlah contoh (*example*) yang cukup. Karena pembangkitan induktif digunakan, maka dibutuhkan *test case* yang cukup untuk membedakan pola yang *valid* dari peluang suatu kejadian.

2.1.2. Pohon Keputusan C4.5

Algoritma *Classification version 4.5* atau biasanya disebut C4.5 adalah pengembangan dari algoritma ID3. Oleh karena pengembangan tersebut, algoritma C4.5 mempunyai prinsip dasar kerja yang sama dengan algoritma ID3. Menurut Khoonsari, et al. (2012), C4.5 memiliki beberapa kelebihan yaitu:

a) C4.5 dapat menangani atribut kontinu dan diskrit.
b) C4.5 dapat menangani *training data* dengan *missing value*.
c) Hasil pohon keputusan C4.5 akan dipangkas setelah dibentuk.
d) Pemilihan atribut yang dilakukan dengan menggunakan *gain ratio*.

Information gain pada ID3 lebih mengutamakan pengujian yang menghasilkan banyak keluaran. Dengan kata lain, atribut yang memiliki banyak nilai yang dipilih sebagai *splitting* atribut. Sebagai contoh, pembagian terhadap atribut yang berfungsi sebagai *unique identifier*, seperti product_ID¸ akan menghasilkan keluaran dalam jumlah yang banyak, di mana setiap keluaran hanya terdiri dari satu *tuple*. Partisi semacam ini tentu saja bersifat *pure*, sehingga informasi yang dibutuhkan untuk mengklasifikasi D berdasarkan partisi seperti ini sebesar *Infoproduct_ID(D)=0*. Sebagai akibatnya, *information gain* yang dimiliki atribut *product_ID* menjadi maksimal. Padahal, jelas sekali terlihat bahwa partisi semacam ini tidaklah berguna. Oleh karena itu, algoritma C4.5 yang merupakan suksesor dari ID3 menggunakan *gain ratio* untuk memperbaiki *information gain*, dengan rumus *gain ratio* adalah:

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

dimana:

S = ruang (data) sampel yang digunakan untuk *training*

A = atribut

Gain(S,A) = *information gain* pada atribut A

SplitInfo(S,A) = *split information* pada atribut A

Atribut dengan nilai *gain ratio* tertinggi dipilih sebagai *atribut test* untuk simpul. Pendekatan ini menerapkan normalisasi pada *information gain* dengan menggunakan apa yang disebut sebagai *split information*. *SplitInfo* menyatakan *entropy* atau informasi potensial dengan rumus:

$$SplitInfo(S, A) = -\sum_{i=1}^{c} \frac{S}{S} log_2 \frac{S_i}{S} \quad \ldots (2.4)$$

dimana:

S = ruang (data) sampel yang digunakan untuk *training*

A = atribut

S_i = jumlah sampel untuk atribut i

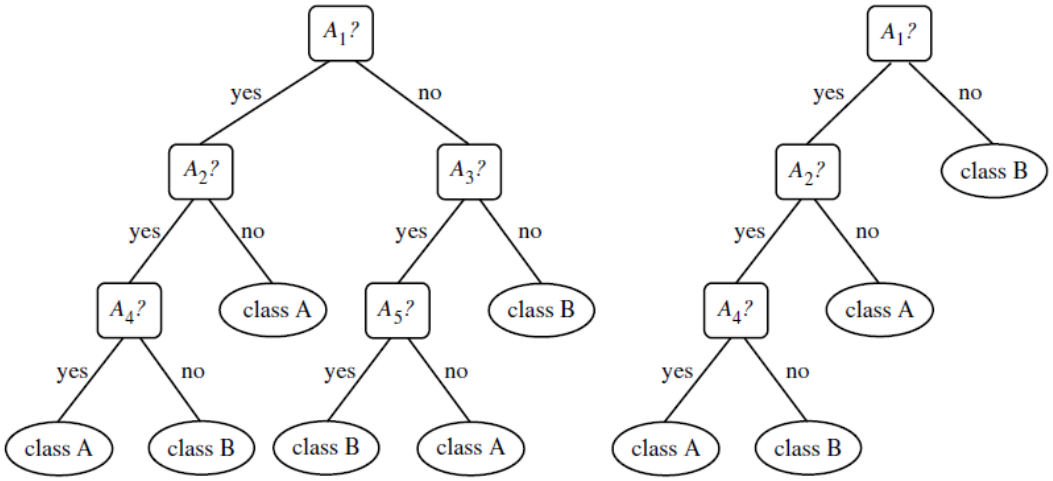
Pada saat pembangunan pohon keputusan, banyaknya cabang mungkin mencerminkan adanya *noise* atau *outlier* pada *training data*. Pemangkasan pohon dapat dilakukan untuk mengenali dan menghapus cabang-cabang tersebut. Pohon yang dipangkas akan menjadi lebih kecil dan lebih mudah dipahami. Pohon semacam itu biasanya juga menjadi lebih cepat dan lebih baik dalam melakukan klasifikasi.

Ada dua metode pemangkasan pohon keputusan, yaitu:

a. *Prepruning* yaitu menghentikan pembangunan suatu *subtree* lebih awal, yaitu dengan memutuskan untuk tidak lebih jauh mempartisi *data training*. Pada pendekatan *prepruning*, sebuah pohon dipangkas dengan cara menghentikan pembangunannya jika partisi yang akan dibuat dianggap tidak signifikan.

b. *Postpruning* yaitu menyederhanakan pohon dengan cara membuang beberapa cabang *subtree* setelah pohon selesai dibangun. Metode *postpruning* ini merupakan metode standar untuk algoritma C4.5.

Gambar 2.3 memperlihatkan perbedaan pohon keputusan sebelum dan setelah dipangkas.

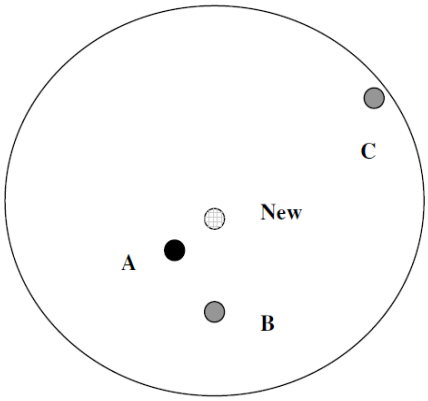


Gambar 2.3. Pohon Keputusan Sebelum dan Setelah Dipangkas (Han, et al., 2006)

Pemangkasan pohon juga dapat digunakan untuk mengatasi *overfitting*. *Overfitting* terjadi karena ada *noise data training*, yaitu data yang tidak relevan sehingga mengakibatkan pohon memiliki *subtree* yang panjang dan tidak seimbang. Misalnya, *internal node* memiliki kelas YA = 5 dan TIDAK = 1. Data yang berada pada kelas TIDAK merupakan *noise*, sehingga apabila data tersebut diolah akan menghasilkan pohon dengan *subtree* yang panjang. *Overfitting* juga dapat terjadi karena *data training* yang sedikit.

2.2. Algoritma k-Nearest Neighbor (k-NN)

Nearest Neighbor adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama, yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada. Misalkan, diinginkan untuk mencari solusi terhadap seorang pasien baru dengan menggunakan solusi dari pasien terdahulu. Untuk mencari kasus pasien mana yang akan digunakan maka dihitung kedekatan kasus pasien baru dengan semua kasus pasien lama. Kasus pasien lama dengan jarak terdekat yang akan diambil solusinya untuk digunakan pada kasus pasien baru.



Gambar 2.4. Ilustrasi Kedekatan Kasus Pasien (Larose, 2005)

Seperti tampak pada Gambar 2.4, terdapat tiga pasien lama A, B, dan C. Ketika ada pasien baru (*New*), maka solusi yang akan diambil adalah solusi dari pasien terdekat dari pasien baru. Seandainya d1 adalah kedekatan antara pasien baru dan pasien A, d2 adalah kedekatan antara pasien baru dengan pasien B, dan d3 adalah kedekatan antara pasien baru dengan pasien C, maka terlihat bahwa d1 lebih dekat daripada d2 dan d3. Dengan demikian, solusi dari pasien A yang akan digunakan untuk memberikan solusi pada pasien baru.

Algoritma k-NN merupakan *distance-based algorithm* yang menggunakan prinsip *similarity* untuk menyelesaikan kasus klasifikasi. Algoritma k-NN sangat sederhana. Algoritma ini bekerja berdasarkan jarak minimum dari data baru terhadap K tetangga terdekat yang telah ditetapkan. Setelah diperoleh K tetangga terdekat, prediksi kelas dari data baru, akan ditentukan berdasarkan mayoritas K tetangga terdekat. (Dunham, 2003)

Adapun penjelasan cara kerja algoritma k-NN adalah:

1. Tentukan parameter k sebagai jumlah tetangga terdekat.
2. Hitung jarak antara data baru dengan semua *data training* dengan rumus *Euclidean Distance* sebagai berikut.

$$dis(t_i, t_j) = \sqrt{\sum_{h=1}^{k} (t_{ih} - t_{jh})^2} \quad \ldots (2.5)$$

   dimana: ti = $\{t_{i1}, \ldots, t_{ik}\}$ dan tj = $\{t_{j1}, \ldots, t_{jk}\}$
3. Urutkan jarak tersebut dan tetapkan tetangga terdekat berdasarkan jarak minimum ke-k.

4. Periksa kelas dari tetangga terdekat.
5. Gunakan mayoritas sederhana dari kelas tetangga terdekat sebagai nilai prediksi data baru.

Berikut ini ditunjukkan algoritma k-NN dalam bentuk *pseudocode* (Dunham, 2003).

Input:

T // Training data

K // Number of neighbors

T // Input tuple to classify

Output:

C // Class to which t is assigned

k-NN algorithm:

// Algorithm to classify tuple using k-NN

N=0

// Find set of neighbors, N, for t

for each $d \in T$ ***do***

if $|N| \leq K$, ***then***

$N = N \cup \{d\};$

else

if $\exists\ u \in N$ *such that* $sim(t,u) \leq sim(t,d)$, ***then***

begin

$N = N - \{u\};$

$N = N \cup \{d\};$

end

// Find class for classification

c = class to which the most $u \in N$ *are classified;*

2.3. Data Mining

Data mining adalah suatu istilah yang digunakan untuk menemukan pengetahuan yang tersembunyi di dalam *database* atau bisa disebut dengan *Knowledge Discovery in Database (KDD). Data mining* merupakan proses semi otomatik yang menggunakan

teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi pengetahuan potensial dan berguna yang bermanfaat yang tersimpan di dalam *database* besar. (Turban, et al., 2005). Menurut *Gartner Group*, *data mining* adalah suatu proses menemukan hubungan yang berarti, pola, dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika (Larose, 2005).

Secara umum, terdapat enam fungsi dalam *data mining* untuk menemukan atau menggali pengetahuan dalam data yang berukuran banyak, yaitu fungsi deskripsi, estimasi, prediksi, klasifikasi, *clustering*, dan asosiasi. Ketiga fungsi pertama dikenal sebagai fungsi minor (fungsi tambahan) dan ketiga fungsi terakhir dikenal sebagai fungsi mayor (fungsi utama). Setiap fungsionalitas akan menghasilkan pengetahuan atau pola yang berbeda satu sama lain dan masing-masing fungsionalitas memiliki algoritma yang dapat diterapkan.

Ada beberapa macam definisi *data mining* menurut para ahli, diantaranya:

1. Proses penemuan pola yang menarik dari data yang tersimpan dalam jumlah besar.
2. Ekstraksi suatu informasi yang berguna atau menarik berupa pola atau pengetahuan dari data yang disimpan dalam jumlah besar.
3. Serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak dikenal, tidak diketahui secara manual dari suatu basis data.
4. *Data mining* adalah proses penemuan informasi yang berguna pada penyimpanan data yang besar secara otomatis. (Tan, et al., 2006)
5. "*Data mining is a process of discovering various models, summaries, and derived values from a given collection of data*." (Mehmed Kantardzic, 2003)

Dari pengertian di atas, dapat disimpulkan unsur-unsur definisi *data mining*, yaitu:

1. Data dalam jumlah besar (*input* pada *data mining*)
2. Eksplorasi dan analisa (proses pada *data mining*)
3. Ekstraksi suatu informasi atau nilai tambah (proses pada *data mining*)
4. Penemuan pola dan pengetahuan (*output* pada *data mining*)

Secara garis besar *data mining* dapat dikelompokkan menjadi 2 kategori utama, yaitu (Tan, et al., 2005):

1. *Descriptive mining*, yaitu proses untuk menemukan karakteristik penting dari data dalam suatu basis data. Teknik *data mining* yang termasuk dalam *descriptive mining* adalah *clustering*, *association*, dan *sequential mining*.
2. *Predictive mining*, yaitu proses untuk menemukan pola dari data dengan menggunakan beberapa variabel lain di masa depan. Salah satu teknik yang terdapat dalam *predictive mining* adalah klasifikasi.

Menurut Han, et al. (2006), istilah lain yang sering dijumpai dan memiliki makna identik dengan *data mining* adalah:

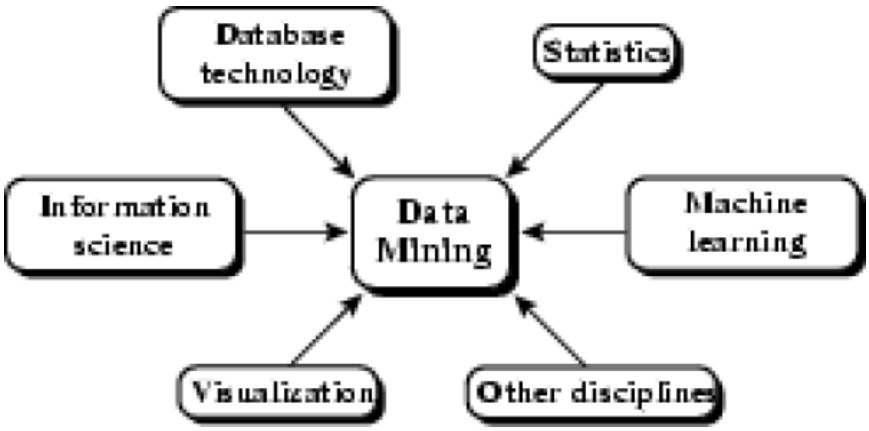
a) *Knowledge Discovery in Databases (KDD)*,
b) *Knowledge Mining from Data*,
c) *Knowledge Extraction*,
d) *Data/Pattern Analysis*,
e) *Data Archaeology*, dan
f) *Data Dredging*.

Adapun yang menjadi latar belakang diperlukannya *data mining* pada era masa kini adalah:

1. Data telah mencapai jumlah dan ukuran yang sangat besar;
    a. Data *web*, *e-commerce*
    b. Data pembelian di toko-toko dan supermarket
    c. Transaksi Bank dan Kartu Kredit
    d. Detail panggilan telepon
    e. Statistik pemerintahan
    f. Rekam medis
    g. *Database* molekul
    h. Citra astronomis
2. Komputer sekarang murah dan *powerful*;
3. Tekanan kompetisi semakin kuat;
4. Menyediakan layanan yang lebih baik dan *customized* (misalnya, dalam *Customer Relationship Management*);
5. Teknik penyimpanan data semakin tinggi dengan kecepatan tinggi (GB/jam);
6. Proses ekstraksi informasi di dalam kumpulan data yang besar menjadi tantangan;

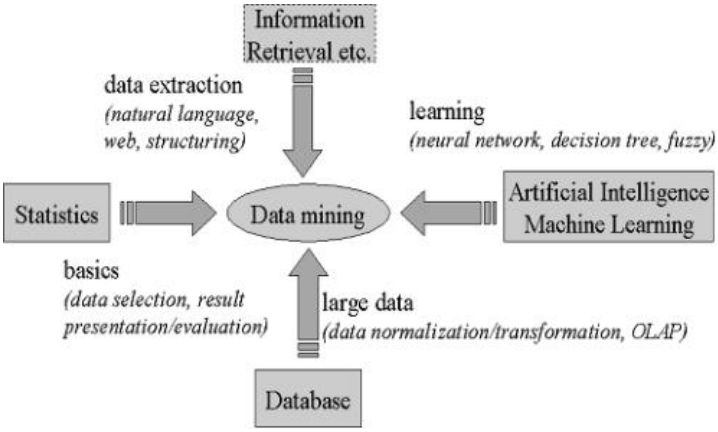
7. Pendekatan analisa data tradisional tidak dapat digunakan utk mengekstrak data dalam jumlah besar;
8. *Data mining* merupakan teknik yang menggabungkan teknik analisa data tradisional dengan algoritma yang *sophisticated* utk pemrosesan data dalam jumlah besar.

Data mining merupakan gabungan dari beberapa bidang ilmu pada bidang Matematika dan Komputer, seperti terlihat pada Gambar 2.5 berikut.



Gambar 2.5. Ilmu *Data Mining* (Han, et al., 2006)

Adapun peran beberapa bidang ilmu terhadap *data mining* ditunjukkan pada Gambar 2.6 berikut.

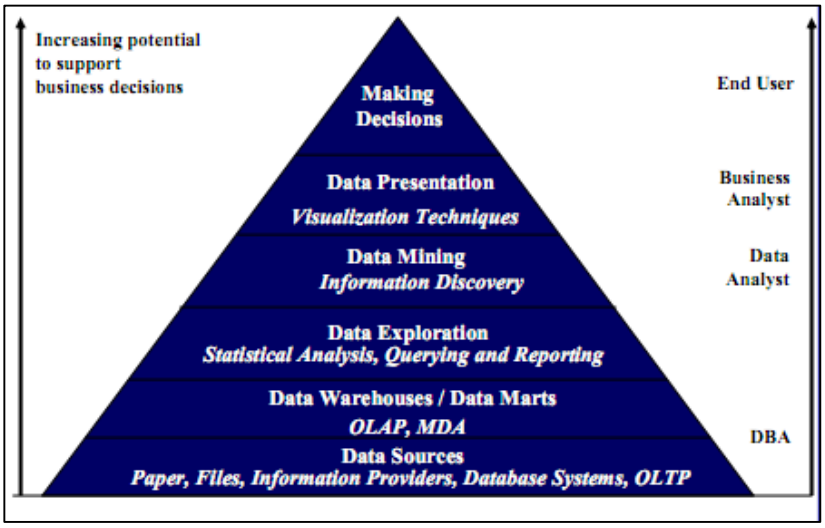


Gambar 2.6. Peranan Bidang Ilmu Lain terhadap *Data Mining*

Istilah lain yang identik dengan *data mining* adalah *Knowledge Discovery in Databases (KDD)*. Adapun *data mining* merupakan salah satu rangkaian dari KDD. Definisi KDD adalah keseluruhan proses non-trivial untuk mencari dan

mengidentifikasi pola (*pattern*) dalam data, dimana pola yang ditemukan bersifat sah, baru, dapat bermanfaat, dan dapat dimengerti.

Adapun hubungan *data mining* dengan *data warehouse* dan *OLAP (On-line Analytical Processing)* adalah teknologi yang ada di *data warehouse* dan *OLAP* dimanfaatkan penuh untuk melakukan *data mining*. Gambar 2.7 menunjukkan posisi masing-masing teknologi dalam pengambilan keputusan.



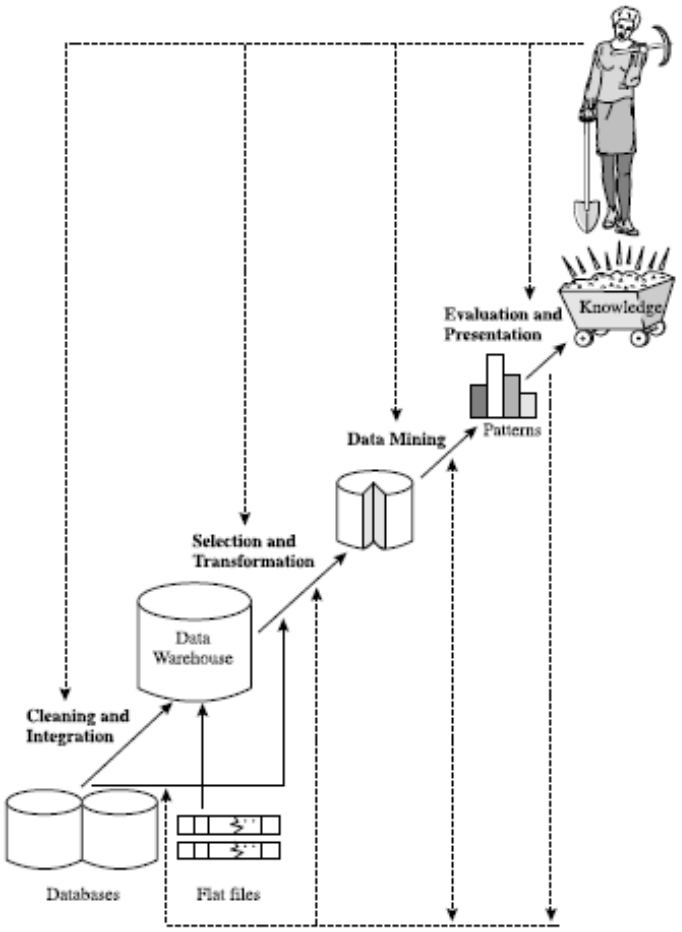
Gambar 2.7. Tingkatan Pemanfaatan Data untuk Pengambilan Keputusan

Dari Gambar 2.7, terlihat bahwa teknologi *data warehouse* digunakan untuk melakukan *OLAP*, sedangkan *data mining* digunakan untuk melakukan *information discovery* yang informasinya lebih ditujukan untuk seorang *Data Analyst* dan *Business Analyst* (dengan ditambah visualisasi).

Dengan memadukan teknologi *OLAP* dengan *data mining*, diharapkan pengguna dapat melakukan hal-hal yang biasa dilakukan di *OLAP* seperti *drilling/rolling* untuk melihat data lebih dalam atau lebih umum, *pivoting*, *slicing*, dan *dicing*. Semua hal tersebut diharapkan nantinya dapat dilakukan secara interaktif dan dilengkapi dengan visualisasi.

Data mining tidak hanya melakukan *mining* terhadap data transaksi saja. Penelitian di bidang *data mining* saat ini sudah merambah ke sistem *database* lanjut seperti *object oriented database*, *image/spatial database*, *time-series data/temporal database*, teks (dikenal dengan nama *text mining*), *web* (dikenal dengan nama *web mining*) dan *multimedia database*.

Adapun langkah-langkah *data mining* dapat dilihat pada Gambar 2.8 berikut.



Gambar 2.8. Tahapan KDD pada *Data Mining* (Han, et al., 2006)

Menurut Han, et al. (2006), KDD meliputi *iterative sequence* sebagai berikut.

- *Data Cleaning*

  Pada tahap ini, dilakukan kegiatan membuang data yang tidak konsisten dan bersifat *noise* dari data yang terdapat di berbagai basis data yang mungkin berbeda format maupun *platform*.

  Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (tipografi).

- *Data Integration*

  Menyatukan sumber data yang tersebar/terpecah sehingga dikumpulkan menjadi *data warehouse*.

- *Data Selection*

  Data yang ada dalam *data warehouse* kemudian direduksi untuk mendapatkan hasil yang akurat, yaitu data yang relevan dengan tugas analisa dikembalikan ke dalam *database*.

  Metode seleksi pada *data mining*

  a. *Sampling*, adalah seleksi subset representatif dari populasi data yang besar.
  b. *Denoising*, adalah proses menghilangkan noise dari data yang akan ditransformasikan
  c. *Feature extraction*, adalah proses membuka spesifikasi data yang signifikan dalam konteks tertentu.

- *Data Transformation*

  Data berubah atau bersatu menjadi bentuk yang paling tepat untuk *mining* dengan ringkasan performa atau operasi.

  Metode transformasi pada *data mining*

  a. *Centering*, mengurangi setiap data dengan rata-rata dari setiap atribut yang ada.
  b. *Normalization*, membagi setiap data yang di-*center*-ing dengan standar deviasi dari atribut bersangkutan.
  c. *Scaling*, mengubah data sehingga berada dalam skala tertentu.

- *Knowledge Discovery* (*Data Mining*)

  Tahapan ini merupakan proses esensial dimana metode yang intelijen (algoritma *data mining*) digunakan untuk mengekstrak pola data. Data yang telah ditransformasi, kemudian ditambang dengan berbagai teknik. Proses *data mining* adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan fungsi-fungsi tertentu. Fungsi atau algoritma dalam *data mining* sangat bervariasi, dimana pemilihannya bergantung pada tujuan dan proses pencarian pengetahuan secara menyeluruh.

- *Pattern Evaluation*

  Untuk mengidentifikasi pola yang benar-benar menarik (*interestingness measures*) yang mewakili pengetahuan berdasarkan atas beberapa tindakan

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

- *Knowledge Presentation (Graphical User Interface)*

  Tahap ini merupakan gambaran teknik visualisasi dan pengetahuan yang digunakan untuk memberikan pengetahuan yang telah ditambang kepada *user*. Pada tahap ini, dipresentasikan pengetahuan dalam bentuk yang mudah dipahami pengguna atau pihak yang berkepentingan.

2.4. Jenis Data dalam Data Mining

Data mining adalah pencarian terhadap pengetahuan/informasi yang bersifat baru, bernilai, dan non-trivial (tidak sederhana) dalam volume data yang besar. Menurut Kantardzic (2003), tugas utama *data mining* dapat diklasifikasikan ke dalam dua kategori, yaitu:

1. Prediktif

   Tujuan dari tugas prediktif adalah untuk memprediksi nilai dari variabel tertentu berdasarkan nilai dari variabel-variabel pada *data sets*. Dalam hal ini, tugas prediktif menghasilkan model sistem berdasarkan *data sets* yang diberikan.

2. Deskriptif

   Tujuan dari tugas deskriptif adalah untuk menemukan pola-pola yang dapat diinterpretasikan oleh manusia. Tugas deskriptif menghasilkan informasi non-trivial dan bersifat baru berdasarkan kepada *data sets* yang tersedia.

Menurut Han, et al. (2006), ada sejumlah *data repositories* yang memungkinkan dilakukannya *data mining*, seperti *relational databases, data warehouses, transactional databases, advanced database systems, flat files, data streams*, dan *World Wide Web (WWW)*.

Adapun penjelasan beberapa jenis data dalam *data mining*, yaitu:

1. *Relational Databases*

Model relasional adalah model data yang paling banyak digunakan saat ini. Pembahasan pokok pada model ini adalah relasi, yang dimisalkan sebagai himpunan dari *record*. Deskripsi data dalam istilah model data disebut skema. Pada model

relasional, skema untuk relasi ditentukan oleh nama, nama dari tiap *field* (atribut atau kolom), dan tipe dari tiap *field*. *Relational Database* disebut juga *database management system (DBMS).*

DBMS adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, misalnya penyimpanan data dalam *field* dan menulis kode aplikasi yang spesifik untuk pengaturannya.

Komponen utama DBMS dapat dibagi menjadi 4 macam, yaitu :

a) Perangkat Keras
b) Perangkat Lunak
c) Data
d) Pengguna

Pengunaan DBMS untuk mengelola data mempunyai beberapa keuntungan, sebagai berikut.

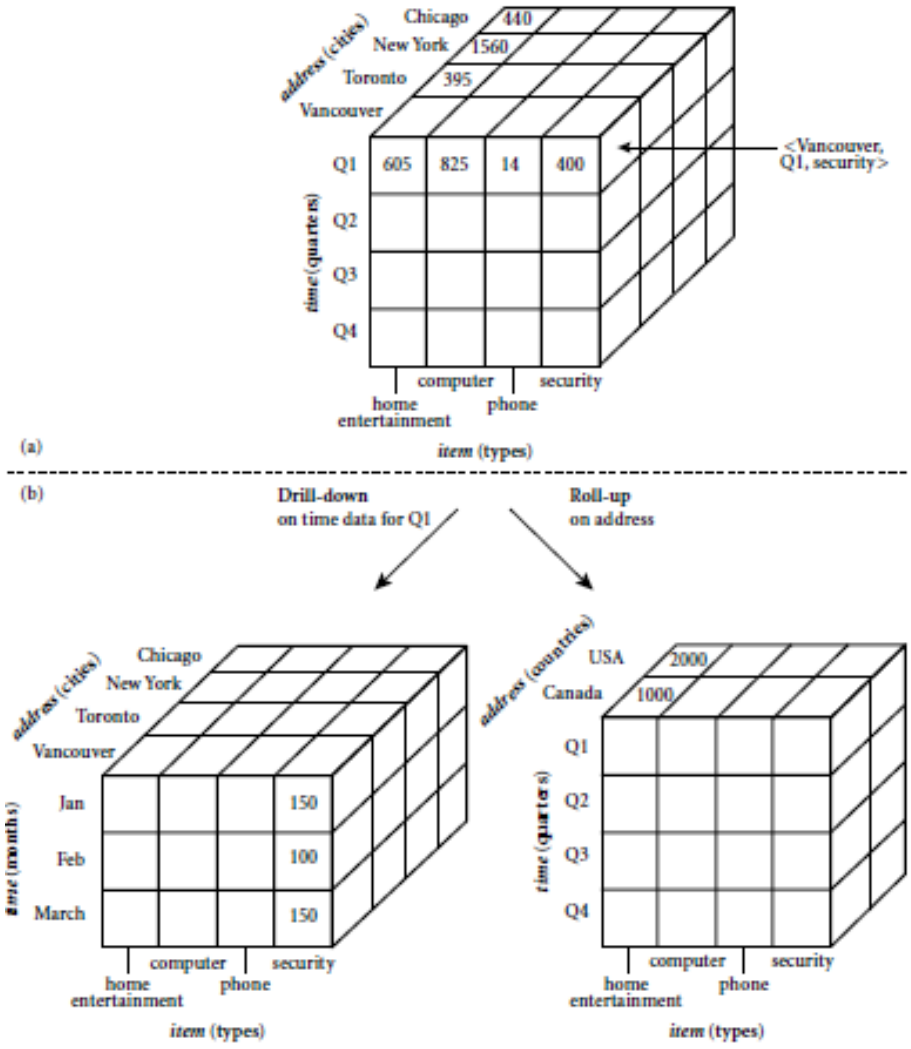
a) Kebebasan data dan akses yang efisien
b) Mereduksi waktu pengembangan aplikasi
c) Integritas dan keamanan data
d) Administrasi keseragaman data
e) Akses bersamaan dan perbaikan dari terjadinya *crash* (tabrakan dari proses serentak)

2. *Data Warehouse*

Biasanya perusahaan-perusahaan memakai *database* dalam operasi sehari-harinya seperti pencatatan transaksi jual-beli, administrasi pengiriman barang, inventori, penggajian, dan sebagainya yang lazim disebut dengan *OLTP* (*online transaction processing*). Dengan makin besarnya kebutuhan akan analisa data untuk mempertahankan keunggulan dalam kompetisi, banyak perusahaan yang juga membangun *database* tersendiri yang khusus digunakan untuk menunjang proses pengambilan keputusan (*decision making*) atau lazim juga disebut dengan *OLAP* (*online analytical processing*).

Berbeda dengan *OLTP* yang hanya memakai operasi *query* yang sederhana dan berulang-ulang, *query* untuk *OLAP* biasanya lebih rumit, bersifat *ad-hoc*, dan tidak

melibatkan operasi data *update*. *OLAP* juga tidak memakai data operasi sehari-hari begitu saja, tetapi memakai data yang sudah terangkum dengan model data yang disebut *data cube*. *Data cube* adalah presentasi data multidimensi seperti jenis barang, waktu, lokasi dan sebagainya. Dimensi *data cube* diperlihatkan pada Gambar 2.9 berikut ini.



Gambar 2.9. *Data Cube* pada *Data Warehouse* (Han, et al., 2006)

Dimensi pada *data cube* dapat dibuat bertingkat, contohnya dimensi lokasi dapat dibagi menjadi kota, provinsi dan negara. Sedangkan dimensi waktu mencakup jam, hari, minggu, bulan, tahun, dan sebagainya. Dengan ini pemakai dapat dengan mudah mendapat rangkuman informasi dari tingkatan dimensi yang lebih luas/umum seperti negara atau tahun dengan operasi yang disebut *roll-up* seperti ditunjukkan di

Gambar 2.9. Sebaliknya dengan operasi *drill-down*, pemakai dapat menggali informasi dari tingkatan dimensi yang lebih detil seperti data harian atau data di lokasi yang spesifik.

Data cube yang tersedia pada *data warehouse* memungkinkan pemakai untuk menganalisa data operasi sehari-hari dengan berbagai sudut pandang, dan sangat berguna untuk mengevaluasi suatu asumsi bisnis. Akan tetapi untuk mendapatkan informasi yang tidak diketahui secara eksplisit diperlukan satu tahap lagi, yaitu teknik *data mining*. Di sini, *data warehouse* merupakan data mentah untuk *data mining*. *Data warehouse* sendiri secara periodik diisi data dari *OLTP* setelah menjalani pembersihan dan integrasi data. Karena itu, ada pula anggapan bahwa *data mining* adalah tahap lanjut dari *OLAP*.

3. *Transactional Databases*

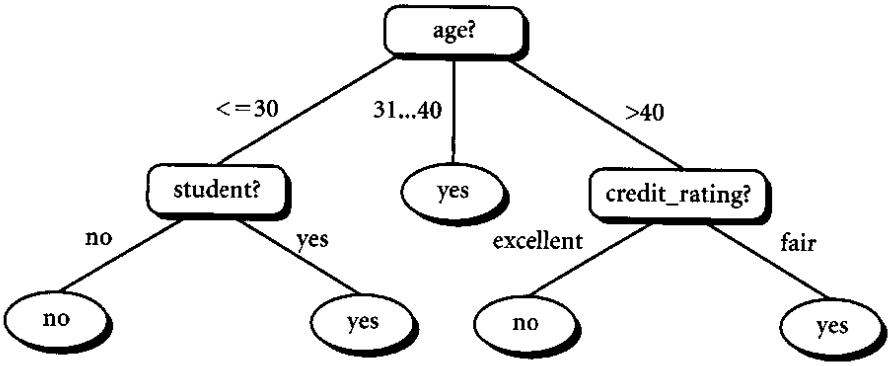
Secara umum, *transactional database* terdiri dari sebuah file dimana setiap *record* merepresentasikan transaksi. Sebuah transaksi biasanya meliputi bilangan identitas transaksi yang unik (trans_id), dan sebuah daftar dari item transaksi (seperti item yang dibeli dalam sebuah stok). *transactional database* dapat memiliki tabel tambahan, yang mengandung informasi lain berkaitan dengan penjualan seperti tanggal transaksi, *customer ID number*, *ID number* dari *sales person* dan dari kantor cabang (*branch*) dimana penjualan terjadi.

Adapun jenis *data sets* pada data mining adalah *private data sets* dan *public data sets*. *Private data sets* merupakan *data sets* yang diambil dari organisasi yang dijadikan objek penelitian, misalnya data bank, rumah sakit, industri, pabrik, dan perusahaan jasa. *Public data sets* merupakan *data sets* yang diambil dari repositori publik yang disepakati oleh para peneliti *data mining*, misalnya *UCI Repository* yang beralamat di *http://archive.ics.uci.edu/ml/datasets.html*. (Bramer, 2007)

Trend penelitian *data mining* saat ini adalah menguji metode menggunakan *public data sets*, sehingga penelitian dapat bersifat: *comparable*, *repeatable*, dan *verifiable*. Untuk kepentingan penelitian tesis ini, penulis menggunakan sumber data berupa *public data sets* yang diambil dari *UCI Repository*.

2.5. Teknik Klasifikasi

Klasifikasi adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Model itu sendiri bisa berupa aturan "jika-maka", berupa pohon keputusan, formula matematis atau *neural network*. Contoh model klasifikasi berupa pohon keputusan pembelian unit komputer ditunjukkan oleh Gambar 2.10.



Gambar 2.10. Contoh Model Klasifikasi (Han, et al., 2006)

Proses klasifikasi biasanya dibagi menjadi dua fase, yaitu *learning* dan *test*. Pada fase *learning*, sebagian data yang telah diketahui kelas datanya diumpankan untuk membentuk model perkiraan. Kemudian pada fase *test*, model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Bila akurasinya mencukupi, model ini dapat dipakai untuk prediksi kelas data yang belum diketahui.

Sebuah teknik klasifikasi adalah pendekatan yang sistematik untuk membuat klasifikasi model dari kumpulan *input data*.

Beberapa teknik klasifikasi yang sering digunakan yaitu, *rule based classsifier*, pohon keputusan, *naive bayes classifier*, *neural network*, dan *fuzzy logiz*. Pengertian untuk masing-masing teknik klasifikasi adalah:

1. Aturan klasifikasi (*Rule-Based Classifier*) adalah suatu bentuk model untuk klasifikasi dalam bentuk IF-THEN.

2. Pohon Keputusan (*Decision Tree*)

   Pohon keputusan merupakan suatu struktur pohon, dimana komponen-komponennya adalah:

   a) *Node*, untuk menunjukkan pengujian terhadap sebuah nilai atribut.
   b) Cabang, untuk merepresentasikan hasil dari pengujian tersebut.
   c) *Leaf*, untuk menunjukkan distribusi kelas.

   Untuk mengklasifikasikan suatu *record* yang belum diketahui kelasnya, nilai-nilai atribut dari *record* tersebut diuji dengan menggunakan pohon keputusan. Penelusuran dilakukan mulai dari akar hingga *node* daun, yang berisi hasil prediksi dari *record* tersebut. Pohon keputusan dapat direpresentasikan dalam bentuk klasifikasi *rule*.

3. Klasifikasi *Bayesian* (*Naive Bayes Classifier*)

   Klasifikasi *Bayesian* menggunakan metode statistik dan menerapkan teorema *Bayes*, dan mampu melakukan prediksi probabilitas keanggotaan.

4. Jaringan Syaraf Tiruan (*Neural Network*)

   Jaringan syaraf tiruan merupakan sekumpulan *node* pemrosesan atau unit yang memetakan *input-output*, dimana antar *node* dihubungkan dengan *link* yang memiliki bobot. Selama proses pelatihan, jaringan melakukan pembelajaran dengan mengatur nilai bobot, sehingga dapat memprediksi nilai suatu kelas dengan benar.

5. *Fuzzy Logic*

   Fuzzy logic adalah suatu sistem yang memungkinkan adanya batasan yang tidak kaku (kabur) untuk mendefinisikan suatu data. Pada himpunan *fuzzy*, terdapat nilai keanggotaan antara 0 hingga 1 yang menunjukkan ukuran terhadap suatu atribut.

Pada teknik klasifikasi, dilakukan analisa terhadap data latih (*training set*) untuk mendapatkan model. Sedangkan data uji (*testing set*) digunakan untuk mengetahui tingkat akurasi dari model yang telah dihasilkan. Klasifikasi dapat digunakan untuk memprediksi nama atau nilai kelas dari objek data.

Proses klasifikasi data dapat dibedakan dalam 2 tahap, yaitu:

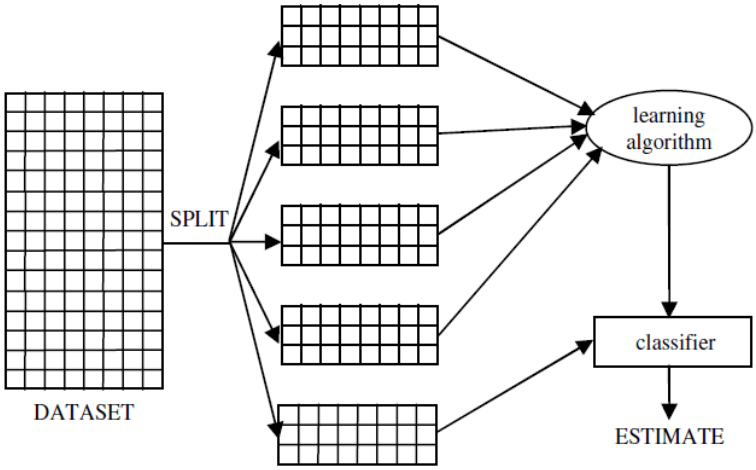
a) Pembelajaran/Pembangunan model

   Setiap *record* pada data latih dianalisa berdasarkan nilai atributnya, dengan menggunakan algoritma klasifikasi untuk mendapatkan model.

b) Klasifikasi

   Pada tahap ini, data uji digunakan untuk mengetahui tingkat akurasi dari model yang dihasilkan. Jika tingkat akurasi yang diperoleh sesuai dengan nilai yang ditentukan, maka model tersebut dapat digunakan untuk mengklasifikasikan *record-record* data baru yang belum pernah dilatihkan atau diujikan sebelumnya.

Adapun metode evaluasi yang digunakan dapat berupa *cross validation*. Pada metode ini, dibentuk k *subset* dari *data sets* yang ada. Misalnya, *5-fold cross validation* berarti 4 *subsets* digunakan sebagai *training sets* dan 1 *subset* digunakan sebagai *testing set*, dengan 5 kali iterasi. Hasil pengukuran adalah nilai rata-rata dari 5 kali pengujian, seperti ditunjukkan pada Gambar 2.11 berikut.



Gambar 2.11. Prosedur *5-fold Cross Validation* (Bramer, 2007)

2.6. Pengukuran Akurasi

Hardikar, et al. (2012) meneliti perbandingan ID3 dan C4.5 dengan parameter *performance* berupa alokasi pemakaian ruang memori (*memory used*), waktu proses yang digunakan (*model build time*), waktu respon model (*search time*), tingkat kesalahan (*error rate*), dan akurasi (*accuracy*).

Adapun *performance* algoritma klasifikasi *data mining* biasanya dinilai dari tingkat akurasi, yaitu persentase *tuples* yang berada pada kelas yang tepat (Dunham, 2003). Misalkan, jika terdapat suatu kelas C_j dan *tuple database* t_i, maka terdapat 4 kuadran kemungkinan penempatan *tuple* t_i ke dalam kelas-kelas yang ada. Hal ini dapat digambarkan sebagai berikut.

TRUE POSITIVE	*FALSE NEGATIVE*
FALSE POSITIVE	*TRUE NEGATIVE*

Gambar 2.12. Hasil Prediksi Kelas (Dunham, 2003)

Keterangan gambar:

- TRUE POSITIVE (TP) berarti t_i diprediksi berada di C_j dan t_i memang seharusnya berada di C_j.
- FALSE POSITIVE (FP) berarti t_i diprediksi berada di C_j, tetapi t_i tidak seharusnya berada di C_j.
- TRUE NEGATIVE (TN) berarti t_i diprediksi tidak berada di C_j dan t_i memang tidak seharusnya berada di C_j.
- FALSE NEGATIVE (FN) berarti t_i diprediksi tidak berada di C_j, tetapi t_i seharusnya berada di C_j.

Aspek akurasi algoritma klasifikasi juga dapat dilihat dari *confusion matrix*. Menurut Han, et al. (2006), *confusion matrix* adalah *tool* yang berguna untuk menganalisa tingkat efektivitas *classifier* dalam mengenali *tuples* dari kelas yang berbeda. Jika terdapat sejumlah kelas m, maka *confusion matrix* merupakan matriks m x m dimana *entry* $c_{i,j}$ menunjukkan menunjukkan jumlah *tuple* dari *data sets* yang dimasukkan ke kelas C_j, padahal kelas sebenarnya adalah C_i. Jelasnya, penilaian terbaik terjadi jika dan hanya jika terdapat nilai nol di luar diagonal matriks. Adapun contoh *confusion matrix* ditunjukkan pada Tabel 2.1 berikut.

Tabel 2.1. *Confusion Matrix*

Actual Membership	*Assignment (Predicted Class)*		
	Short	*Medium*	*Tall*
Short	0	4	0
Medium	0	5	3
Tall	0	1	2

Adapun struktur *table of confusion* ditunjukkan sebagai berikut.

Tabel 2.2. *Table of Confusion*

TRUE POSITIVE (TP)	FALSE NEGATIVE (FN)
FALSE POSITIVE (FP)	TRUE NEGATIVE (TN)

Menurut Witten, et al. (2005), para peneliti *Information Retrieval (IR)* menggunakan parameter berupa *recall*, *precision*, dan *F-measure*. Rumus penentuan *Precision, Recall,* dan *F-Measure* berdasarkan *table of confusion* adalah sebagai berikut.

$$Precision = \frac{TP}{TP + FP} \quad \ldots (2.6)$$

$$Recall = \frac{TP}{TP + FN} \quad \ldots (2.8)$$

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad \ldots (2.7)$$

$$F - measure = \frac{2.TP}{2.TP + FP + FN} \quad \ldots (2.9)$$

Menurut Bramer (2007), *Precision* disebut juga sebagai *Positive Prediction Value*, *Recall* disebut juga sebagai *TP Rate/Hit Rate/Sensitivity*, dan *F-measure* disebut juga sebagai *F1 Score (a harmonic mean of Precision and Recall)*.

Sedangkan aspek pengukuran akurasi yang lain berupa *Predicitive Accuracy* menurut Bramer (2007) ataupun *Success Rate* menurut Witten, et al. (2005) dapat dihitung berdasarkan persamaan berikut.

$$Success\ Rate = \frac{TP + TN}{P + N} \quad \ldots (2.10)$$

dimana: P = TP + FN dan N = FP + TN

2.7. Riset Terkait

Menurut Wu, et al. (2007), terdapat sejumlah algoritma fungsi *data mining* yang dipilih menjadi *Top 10 algorithm data mining*. Hasilnya, diperoleh bahwa algoritma klasifikasi menjadi mayoritas dalam *Top 10 algorithm*. Adapun algoritma klasifikasi yang terbaik dalam *Top 10 algorithm data mining* adalah pohon keputusan C4.5, *k-NN*, *Naive Baiyes*, dan *Classification and Regression Trees (CART).*

Galathiya, et al. (2012) melakukan penelitian dengan melakukan pengembangan pada algoritma pohon keputusan C4.5 dan melakukan perbandingan terhadap algoritma Naive Bayes, *Sequential Minimal Optimization (SMO)*, dan *multilayer perceptron* dengan menggunakan sejumlah *data sets* dan pengujian akurasi berdasarkan kepada *Correctly Classified Instances* dan *Relative Absolute Error* menggunakan metode *10-fold cross validation*. Hasilnya, algoritma pohon keputusan C4.5 yang dikembangkan memiliki tingkat akurasi sekitar 1-3% lebih baik pada 7 *data sets* dari total 9 *data sets*. Penelitian ini belum melibatkan algoritma k-NN dan belum menunjukkan hasil pengukuran akurasi dalam *Confusion Matrix*.

Defiyanti & Pardede (2010) melakukan penelitian dengan melakukan perbandingan kinerja algoritma pohon keputusan *Iterative Dichotomiser 3* (ID3) dan C4.5 dalam klasifikasi *Spam-mail*. Pengukuran kinerja didasarkan atas jumlah data dan jumlah atribut. Hasilnya, diperoleh bahwa algoritma ID3 memiliki kinerja (*precision*, *recall*, dan *accuracy*) yang lebih baik dibandingkan C4.5. Pada penelitian ini, belum digunakan metode *k-fold cross validation* dalam mengukur performa kedua macam algoritma.

Othman & Yau (2007) melakukan penelitian berjudul "*Comparison of Different Classification Techniques using WEKA for Breast Cancer*". Metode pengujian yang digunakan berupa *percentage split* sebesar 75%, artinya 75% data digunakan sebagai *data training* dan 25% sisanya sebagai *data testing*. Pengujian akurasi berdasarkan kepada *Correctly Classified Instances, Incorrectly Classified Instances,* dan *Time Taken*. Hasilnya, algoritma *Naive Bayes* memiliki tingkat akurasi tertinggi dan waktu pembangunan model tercepat dibandingkan dengan *Radial Basis Function, Decision Tree and Pruning, Single Conjunctive Rule Learner,* dan *Nearest Neighbors Algorithm*. Pada penelitian ini, hanya digunakan satu macam *data set* berupa *breast cancer* dan belum menggunakan metode *k-fold Cross Validation*.

2.8. Perbedaan dengan Riset yang Lain

Dalam penelitian ini, dilakukan pelatihan (*training*) dan pengujian (*testing*) terhadap 5 *UCI data sets* menggunakan algoritma klasifikasi pohon keputusan dan k-NN dan menghasilkan *confusion matrix* untuk penilaian *precision*, *recall*, *F-measure*, dan *success rate*. Adapun metode pengujian/evaluasi yang digunakan adalah *10-fold cross validation*.

2.9. Kontribusi Riset

Aspek akurasi yang dihasilkan dalam penelitian ini meliputi *confusion matrix* untuk penilaian *precision*, *recall*, *F-measure*, dan *success rate* (persentase *correctly classified instances*). Dengan demikian, diharapkan dapat diketahui aspek pengukuran *performance* suatu algoritma klasifikasi dari sudut pandang akurasi, serta diperoleh perbandingan akurasi algoritma pohon keputusan dan k-NN pada implementasi terhadap 5 *UCI data sets.*