BAB II

TINJAUAN PUSTAKA

2.1 Data Mining

2.1.1 Pengertian Data Mining

Dengan semakin besarnya jumlah data dan kebutuhan akan analisis data yang akurat maka dibutuhkan metode analisis yang tepat. Data mining merupakan teknik yang menggabungkan teknik analisis data dengan algoritma untuk melakukan analisis data dan menemukan pola-pola penting data.

Secara sederhana, data mining atau penambangan data dapat didefinisikan sebagai proses seleksi, eksplorasi, dan pemodelan dari sejumlah besar data untuk menemukan pola atau kecenderungan yang biasanya tidak disadari keberadaannya [HAN-01]. Data mining dapat dikatakan sebagai proses mengekstrak pengetahuan dari sejumlah besar data yang tersedia [HAN-01]. Pengetahuan yang dihasilkan dari proses data mining harus baru, mudah dimengerti, dan bermanfaat. Dalam data mining, data disimpan secara elektronik dan diproses secara otomatis oleh komputer menggunakan teknik dan perhitungan tertentu.

Alasan-alasan utama dalam penggunaan data mining adalah [HAN-01]:

1. Banyaknya jumlah data yang ada dan akan terus meningkatnya jumlah data.
2. Kebutuhan untuk menginterpretasikan data

Ada beberapa definisi data mining, diantaranya:

1. Data mining adalah disiplin ilmu yang tujuan utamanya adalah untuk menemukan, menggali, atau menambang pangetahuan dari data atau informasi yang kita miliki [1].

2. Data mining adalah suatu proses otomatis terhadap data yang sudah ada, data yang diproses berupa data yang sangat besar [2].
3. Data mining *(knowledge discovery in database)* adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data beukuran besar [3].
4. *Data mining* adalah bagian integral dari *knowledge discovery in databases* (KDD).
5. *Data mining* adalah sebuah proses percarian secara otomatis informasi yang berguna dalam tempat penyimpanan data berukuran besar

Dari beberapa definisi diatas maka dapat disimpulkan bahwa, data mining adalah metode secara otomatis menemukan informasi yang berguna dan tersimpan pada data dengan ukuran yang sangat besar sehingga ditemukan pola menarik yang sebelumnya tidak diketahui. Teknik data mining difungsikan untuk mendapatkan deskripsi dari data dan mendapatkan model dari data yang berguna untuk prediksi. Deskripsi berarti menemukan pola yang mudah dipahami oleh pengguna dalam menggambarkan data, contohnya: *Clustering, Association Rule Discovery, Sequential Pattern Discovery*, sedangkan prediksi berarti menemukan pola untuk memprediksi nilai dari suatu variabel yang nilainya belum diketahui contohnya: *Classification, Regression, Deviation Detection.* Kata mining sendiri berarti usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar.

Ada beberapa karakteristik dari data mining diantaranya:

a. *Data mining* berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
b. *Data mining* biasa menggunakan data yang sangat besar. Biasanya data yang besar digunakan untuk membuat hasil lebih dipercaya.
c. *Data mining* berguna untuk membuat keputusan yang kritis, terutama dalam strategi.

2.1.2 Fungsi –Fungsi Data Mining

Beberapa fungsi dalam data mining, yaitu:

1. Fungsi Prediksi (*prediction*)

   Proses untuk menemukan pola dari data dengan menggunakan beberapa variabel untuk memprediksikan variabel lain yang tidak diketahui jenis atau nilainya.

2. Fungsi Deskripsi (*description*)

   Proses untuk menemukan suatu karakteristik penting dari data dalam suatu basis data.

3. Fungsi Klasifikasi (*classification*)

   Klasifikasi merupakan suatu proses untuk menemukan model atau fungsi untuk menggambarkan *class* atau konsep dari sutau data. Proses yang digunakan untuk mendeskripsikan data yang penting serta dapat meramalkan kecenderungan data pada masa depan.

4. Fungsi Asosiasi (*association*)

   Proses ini digunakan untuk menemukan suatu hubungan yang terdapat pada nilai atribut dari sekumpulan data.

2.1.3 Proses Data Mining

1. Mendefinisikan masalah, menentukan input dan output untuk form, menentukan nilai efisiensi, menentukan keakuratan dan sebagainya.
2. Mengumpulkan dan memilih data yang akan digunakan.
3. Mempersiapkan data, seperti mengubah bentuk data ke bentuk yang sesuai, pembersihan data, atau menggabungkan data dari sumber yang berbeda.
4. Menentukan metode yang sesuai terdiri dari dua bagian:
   a. Memilih model atau algoritma, seprti menggunakan model yang seperti apa, memilih untuk menggunakan algoritma apa
   b. Memilih parameter model, misalnya jumlah node pada tiap tingkat jika artificial network digunakan.
5. Pelatihan atau *testing*, melakukan tes pada data dengan menerapkannya menggunakan algoritma.

6. Interpretasi, evaluasi dan visualisasi pola: adanya sesuatu yang baru dan menarik, lakukan iterasi jika diperlukan.

2.1.4 Pengelompokan Data Mining

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu:

1. Klasifikasi

   Klasifikasi merupakan tugas data mining yang paling umum. Ciri dari klasifikasi adalah memiliki definisi yang jelas tentang kelas-kelas (*predifined classes*) dan *training set*. Klasifikasi bertujuan memprediksi kelas dari suatu data yang belum diketahui kelasnya. Dalam mencapai tujuan tersebut, proses klasifikasi membentuk suatu model yang mampu membedakan data kedalam kelas-kelas yang berbeda berdasarkan aturan atau fungsi tertentu.
2. Deskripsi

   Deskripsi adalah cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data.

3. Estimasi

   Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi.

4. Pengelompokan (*clustering*)

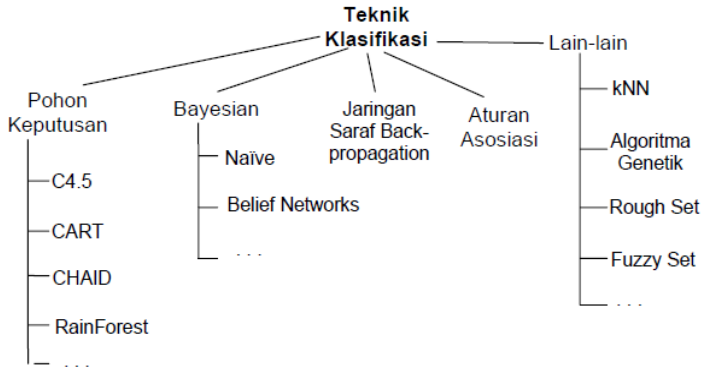
   Pengelompokan adalah tugas data mining yang menggunakan metode populasi yang heterogen menjadi sejumlah kelompok data yang homogen. Data dikelompokan berdasarkan ciri-ciri yang sama tidak tergantung pada *predefined classes* dan training set.

5. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

2.2 Klasifikasi

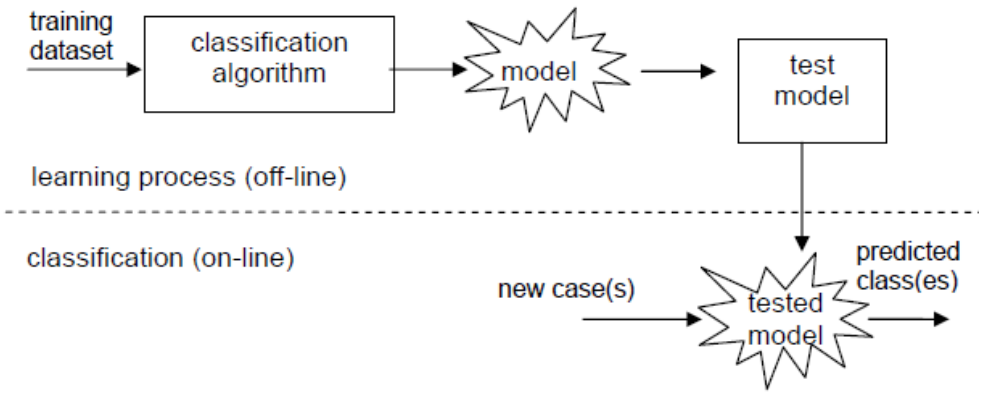
Teknik klasifikasi adalah suatu proses yang menemukan properti-properti yang sama pada sebuah himpunan obyek di dalam sebuah basis data, dan mengKlasifikasikannya ke dalam kelas-kelas yang berbeda menurut model klasifikasi yang ditetapkan. Klasifikasi dalam data mining dikelompokkan ke dalam teknik pohon keputusan, Bayesian (Naïve Bayesian dan Bayesian Belief Networks), Jaringan Saraf Tiruan (Backpropagation), teknik yang berbasis konsep dari penambangan aturan-aturan asosiasi, dan teknik lain (k-Nearest Neighboor, algoritma genetik, teknik dengan pendekatan himpunan rough dan fuzzy). Setiap teknik memiliki kelebihan dan kekurangannya sendiri, berikut gambar pengelompokan teknik klasifikasi.



Gambar 1 Pengelompokan Teknik Klasifikasi

Secara umum, proses klasifikasi dapat dilakukan dalam dua tahap, yaitu proses belajar dari data pelatihan dan klasifikasi kasus baru. Pada proses belajar, algoritma klasifikasi mengolah data pelatihan untuk menghasilkan sebuah model. Setelah model diuji dan dapat diterima, pada tahap klasifikasi, model tersebut digunakan untuk memprediksi kelas dari kasus baru untuk membantu proses

pengambilan keputusan (Han *et al*.,2001; Quinlan, 1993). Kelas yang dapat diprediksi adalah kelas-kelas yang sudah terdefinisi pada data pelatihan. Karena proses klasifikasi kasus baru cukup sederhana, penelitian lebih banyak ditujukan untuk memperbaiki teknik-teknik pada proses belajar.



Gambar 2 Skema Klasifikasi secara Umum

2.2.1 Klasifikasi Dengan Decision Tree

Beberapa contoh algoritma yang mengimplementasikan klasifikasi dengan metode decision tree antara lain:

1. *Hunt's Algorithm*.
2. *ID3 Algorithm* (Algoritma ID3).
3. *C4.5 Algorithm* (Algoritma C4.5).
4. *CART* (*Classification and Regrssion Trees*) *Algorithm* (Algoritma CART).
5. *CHAID* (*Chi-squared Automatic Interaction Detector*) *Algorithm* (Algoritma CHAID).
6. *SLIQ Algorithm*.
7. *SPRINT Algorithm*, dan lain-lain.

Keberhasilan suatu algoritma klasifikasi dan prediksi biasanya ditentukan oleh kriteria-kriteria berikut:

1. Akurasi Prediksi

Kemampuan dalam ketepatan memprediksikan nilai ke kelas yang tepat.

2. Kecepatan

   Kecepatan komputasi dalam menghasilkan suatu model prediksi.

3. Efisiensi

   Kemampuan dalam membuat pemodelan yang efisiensi dengan banyaknya data yang diberikan.

4. Interpretasi

   Kemampuan pemahaman dan pengetahuan yang dapat dipresentasikan. Teknik yang dapat diimplementasikan untuk klasifikasi dan prediksi dalam data mining, antara lain: teknik pembuatan pohon keputusan (*decision tree*), bayesian (*naive bayesian* dan *bayesian belief network*), jaringan sarap tiruan (*back propagation*), dan teknik lainnya (*k-nearest neighboor*, algoritma genetik, pendekatan himpunan *rough* dan *fuzzy*).

Setiap teknik memiliki kelebihan dan kekurangan masing-masing. Beberapa teknik akan sangat efektif untuk diimplementasikan pada kondisi tertentu dan beberapa akan menghasilkan yang sebaliknya. Dengan pemilihan teknik yang tepat, akan didapatkan hasil yang maksimal dari penambangan data.

Didalam data mining, pemodelan tree dapat dideskripsikan kedalam tiga kategori, yakni:

1. *Classification tree*, menganalisis ketika keluaran prediksi merupakan bagian data dari kelas yang dimiliki.
2. *Regression tree*, menganalisis ketika keluaran prediksi dapat berupa angka (seperti: harga murah, lamanya pasien tinggal, dan sebagainya).
3. *Classification and regression Tree*, digunakan ketika keluaran prediksi dapat berupa *numeric* maupun *non-numeric*.

2.3 Pohon Keputusan (*Decision Tree*)

Pohon keputusan adalah struktur *flowcart* yang mempunyai *tree* (pohon), dimana setiap simpul internal menandakan suatu tes atribut. Setiap cabang merepresentasikan kelas atau distribusi kelas, alur pada *decision tree* ditelusuri dari simpul ke akar ke simpul daun yang memegang prediksi kelas untuk contoh tersebut. *Decision tree* mudah untuk dikonversikan keaturan klasifikasi (*classification rule*). Konsep data dalam *decision tree* dinyatakan dalam bentuk tabel dengan atribut dan record.

Decision tree digunakan untuk kasus-kasus yang keluarannya bernilai diskrit. Banyak variasi model *decision tree* dengan tingkat kemampuan dan syarat yang berbeda, pada umumnya beberapa ciri yang cocok untuk diterapkannya *decision tree* adalah sebagai berikut :

1. Data dinyatakan dengan pasangan atribut dan nilainya
2. Label atau keluaran data biasanya bernilai diskrit
3. Data mempunyai *missing value* (nilai dari suatu atribut tidak diketahui).

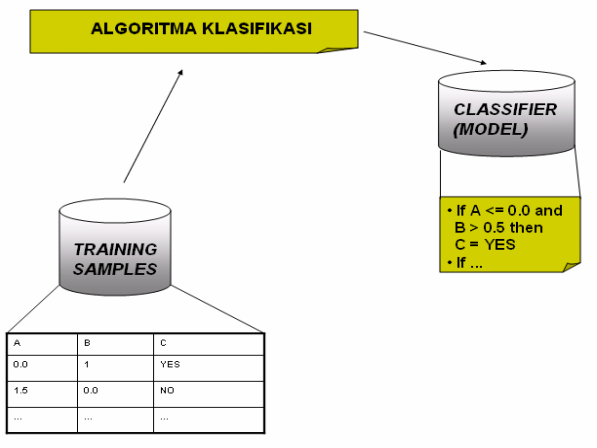
Dengan cara ini akan mudah mengelompokkan obyek kedalam beberapa kelompok. Untuk membuat *decision tree* perlu memperhatikan hal-hal berikut ini:

a. Atribut mana yang akan dipilih untuk pemisahan obyek
b. Urutan atribut mana yang akan dipilih terlebih dahulu
c. Struktur *tree*
d. Kriteria pemberhentian
e. Pruning

2.3.1 Proses Pembentukan Pohon Keputusan (*Decision Tree*)

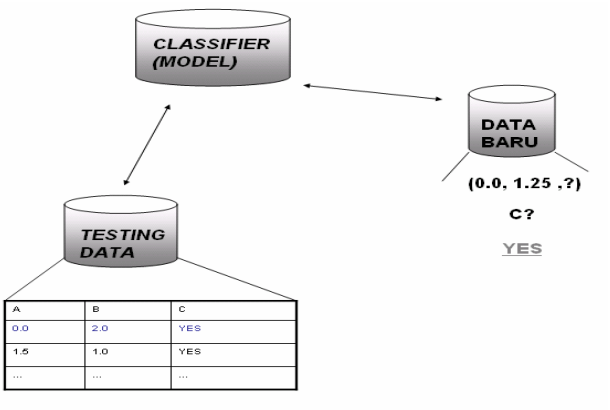
Decision Tree adalah sebuah struktur pohon, dimana setiap node pohon merepresentasikan atribut yang telah diuji, setiap cabang merupakan suatu pembagian hasil uji, dan node daun (*leaf*) merepresentasikan kelompok kelas tertentu. Level node teratas dari sebuah *decision tree* adalah node akar (*root*) yang biasanya berupa atribut yang paling memiliki pengaruh terbesar pada suatu kelas tertentu.

Pembuatan model, pada tahap ini setiap data diasumsikan telah digolongkan kedalam sejumlah kelas (*predefined class*). Himpunan data yang akan menyusun model ini disebut sebagai *training data*. Model yang dihasilkan dipresentasikan dalam bentuk aturan klasifikasi, pohon keputusan atau formula matematika.



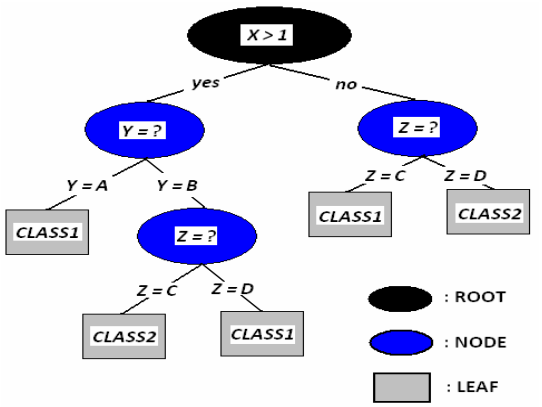
Gambar 3 Ilustrasi Pembuatan Pohon Keputusan

Pemanfaatan model, tahap ini digunakan untuk mengklasifikasikan obyek yang belum diketahui kelasnya. Estimasi akurasi dilakukan dengan membandingkan kelas dari testing data dengan kelas hasil klasifikasi model. Tingkat akurasi adalah ratio jumlah testing data yang diklasifikasikan secara benar berdasarkan model klasifikasi dengan seluruh jumlah testing data. Jika tingkat akurasi ini diterima maka model klasifikasi kemudian dapat digunakan untuk mengklasifikasikan data yang belum diketahui kelasnya.



Gambar 4 Ilustrasi Pemanfaatan Pohon Keputusan

Representasi pohon keputusan ini dianggap sebagai metode logis yang sering digunakan pada bahasan mengenai statistik terapan dan pembelajaran mesin (*machine learning*). Pembuatan pohon keputusan sendri menggunakan metode *supervised learning* yaitu proses pembelajaran dimana data ini terdiri dari nodes atau simpul yang merupakan atribut dari data sampel. Cabang (*branches*) yang keluar dari node tersebut merupakan nilai atau *outcome* yang dimiliki oleh atribut (*nodes*) bersangkutan. Sedangkan daun yang ada pada pohon keputusan tersebut menunjukan kelas dari data sampel yang diuji. Sebagai ilustrasi dapat dilihat pada contoh gambar berikut:



Gambar 5 Model Pohon Keputusan

Pada gambar 5 terlihat ada 3 atribut berbeda yaitu X, Y, dan Z yang terletak pada simpul (*node*) berbentuk oval. Atribut X terletak pada simpul akar

(*root node)* sedangkan Y dan Z terdapat di dalam *internal node* atau simpul dalam. Tiap cabang yang keluar dari simpul tersebut menunjukkan nilai masing-masing atribut yang dimiliki oleh data pengujian. Pada simpul daun (*leaf node)* terdapat kelas yang menjadi keluaran akhir dari *classifier*. Untuk mengetahui kelas dari suatu data pengujian maka jalur yang ada dari akar hingga daun dapat ditelusuri.

Dalam pohon keputusan, leaf node merupakan sebuah label kelas sedangkan non terminal node yang terdiri dari root dan internal node lainnya, mengandung kondisi uji atribut untuk memisahkan record yang memiliki karakteristik berbeda.

Setelah pohon keputusan dikonstruksi, data tes dapat diklasifikasi. Bermula dari root, kondisi tes diaplikasikan ke record dan mengikuti cabang yang sesuai berdasarkan keluaran dari tes.

2.4 Algoritma *ID3*

Iterative Dichotomicer 3 (ID3) adalah algoritma *decision tree learning* (algoritma pembelajaran pohon keputusan) yang paling dasar. Algoritma ini melakukan pencarian secara rakus atau menyeluruh (*greedy*), pada semua kemungkinan pohon keputusan. Salah satu algoritma induksi pohon keputusan yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan. Algoritma ID3 dapat diimplementasikan menggunakan fungsi rekursif (fungsi yang memanggil dirinya sendiri). Algoritma ID3 berusaha membangun *decision tree* (pohon keputusan) secara *top-down* (dari atas kebawah), mulai dengan pertanyaan : “atribut mana yang pertama kali harus dicek dan diletakkan pada *root*?” pertanyaan ini dijawab dengan mengevaluasi semua atribut yang ada dengan menggunakan suatu ukuran statistik (yang banyak digunakan adalah *information gain*) untuk mengukur *efektivitas* suatu atribut dalam mengklasifikasikan kumpulan sampel data. Berikut adalah cara kerja dari algoritma ID3:

1. Pemilihan atribut dengan menggunakan *Information Gain*.

2. Pilih atribut dimana nilai gainnya paling besar.
3. Buat simpul yang berisi atribut tersebut.
4. Proses perhitungan *Information Gain* akan terus dilaksanakan sampai semua data masuk kedalam kelas yang sama, sedangkan atribut yang telah dipilih tidak diikutkan lagi dalam perhitungan nilai *information gain*. *Gain* mengukur seberapa baik suatu atribut memisahkan *training exm ple* kedalam kelas target. Atribut dengan informasi tertinggi akan dipi lih. Dengan tujuan untuk mendefinisikan *gain*, pertama-tama digunakanlah ide dari teori informasi yang disebut *entropi. Entropi* mengukur jumlah dari informasi .

2.4.1 Entropy Information Gain

Sebuah obyek yang diklasifikasikan dalam pohon harus dites nilai entropinya. *Entropy* adalah ukuran dari teori informasi yang dapat mengetahui karakteristik dari *impuryt* ,dan *homogenity* dari kumpulan data. Dari nilai entropy tersebut kemudian dihitung nilai *information gain* (IG) masing-masing atribut.

Rumus menghitung entropi informasi adalah:

$$Entropy(S) = (- p_{+} \log 2 p_{+}) + (-p_{-} \log 2 p_{-}) \quad \text{.....(1)}$$

Keterangan:

S = Himpunan kasus atau 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. Dari rumus entropy diatas dapat disimpulkan bahwa definisi entropy (S) adalah jumlah bit yang diperkirakan dibutuhkan untuk dapat mengekstrak suatu kelas (+ atau -) dari sejumlah data acak pada suatu ruang sampel S. Entropy bisa dikatakan sebagai kebutuhan bit untuk menyatakan suatu kelas. Semakin kecil

nilai dari suatu entropy maka semakin baik digunakan dalam mengekstraksi suatu kelas.

Panjang kode untuk menyatakan informasi secara optimal adalah log2p bits untuk message yang mempunyai probabilitas p. Sehingga jumlah bit yang diperkirakan untuk mengekstraksi S kedalam kelas adalah: = - p_+ $\log_2 p_+$ + -p_- $\log_2 p_-$

2.4.2 Information Gain

Setelah mendapatkan nilai entropy untuk suatu kumpulan data, maka kita dapat mengukur efektivitas suatu atribut dalam mengklasifikasikan data. Ukuran efektifitas ini disebut informasi gain. Secara matematis, informasi gain dari suatu atribut A, dituliskan sebagai berikut:

Gain (S,A)= Entropy(S) - $\sum \frac{|Sv|}{|S|} Entropy(S_V)$ .....(2)

Keterangan :

A = atribut

V = suatu nilai 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

Entropy (S) = 0 jika semua contoh pada S berada dalam kelas yang sama.

Entropy (S) = 1 jika semua jumlah contoh positif dan jumlah negatif dalam S adalah sama.

Entropy (S) = < 1 jika jumlah contoh positif dan negatif dalam S tidak sama

2.4.3 Berikut algoritma dari ID3

Input: sampel training, label training, atribut

1. Membuat simpul akar untuk tree yang dibuat,
2. Jika semua sampel positif, berhenti dengan suatu pohon dengan satu simpul akar, beri label (+),

3. Jika semua sampel negatif, berenti dengan suatu pohon dengan satu simpul akar, beri label (-),
4. Jika atribut kosong, berhenti dalam dengan satu pohon dengan satu simpul akar dengan label sesuai nilai yang terbanyak yang ada pada label training untuk yang lain,

   Mulai

   a. A ⟵ atribut yang mengklasifikasikan sample dengan hasil terbaik (berdasarkan *information gain*),
   b. Atribut keputusanuntuk simpul akar ⟵ A,
   c. Untuk setiap nilai, v_i , yang mungkin untuk A,
   d. Tambahkan cabang dibawah akar yang berhubungan dengan $A = v_{i,}$
   e. Tentukan sampel Sv_i sebagai subset dari sampel yang mempunyai nilai v_i untuk atribut A,
   f. Jika sampel Sv_i kosong, dibawah cabang tambahkan simpul daun dengan label = nilai yang terbanyak yang ada pada label training, yang lain tambah cabang baru dibawah cabang yang sekarang ID3 (sampel training, label training, atribut-[A])

5. Berhenti

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

1. Deskripsi atribut nilai, atribut yang sama harus mendeskripsikan tiap contoh dan memiliki jumlah nilai yang sudah ditentukan.
2. Kelas yang sudah didefinisikan sebelumnya, suatu atribut contoh harus sudah didefinisikan, karena tidak dipelajari oleh *ID3*.
3. Kelas-kelas yang diskrit, kelas harus digambarkan dengan jelas. Kelas yang kontinu dipecah-pecah menjadi kategori-kategori yang relatif.

4. Jumlah contoh (*example*) yang cukup, karena pembangkitan induktif digunakan, maka dibutuhkan *test case* yang cukup untuk membedakan pola yang valid dari peluang suatu kejadian.
5. Pemillihan atribut pada ID3 dilakukan dengan properti statistik, yang disebut dengan information gain. *Gain* mengukur seberapa baik suatu atribut memisahkan *training example* ke dalam kelas target. Atribut dengan informasi tertinggi akan dipilih. Dengan tujuan untuk mendefinisikan *gain,* pertama-tama digunakanlah ide dari teori informasi yang disebut entropi. Entropi mengukur jumlah dari informasi yang ada pada atribut.