# Penerapan Pohon Untuk Machine Learning

**Muhammad Noor Adityana (13506052)**

Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
e-mail: if16052@students.if.itb.ac.id

## ABSTRAK

**Makalah ini membahas salah satu penggunaan pohon dalam bidang informatika, tepatnya pada bidang intelegensia buatan. Penggunaan pohon yang akan dibahas dalam makalah ini adalah dalam hal machine learning. Salah satu metode dari machine learning adalah menggunakan decision tree. Sesuai namanya, decision tree merupakan salah satu aplikasi dari pohon.**

**Kata kunci:** Pohon, Decision Tree, Intelegensia Buatan.

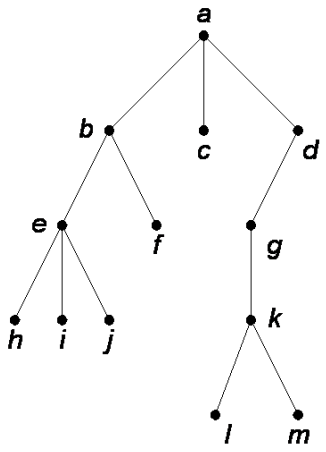
## 1. PENDAHULUAN

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit[1]. Pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah disebut pohon berakar (*rooted tree*). Bentuk dari pohon berakar dapat dilihat pada gambar 1.

Terminologi yang digunakan pada pohon berakar adalah sebagai berikut.

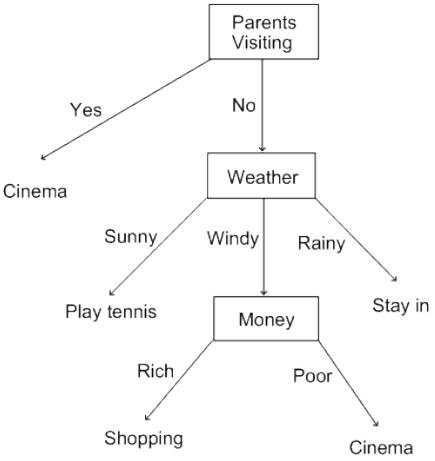
1. **Anak (child atau children) dan Orangtua (*parent*)**
   b, c, dan d adalah anak-anak dari simpul a
   a adalah orang tua dari anak-anak itu
2. **Lintasan**
   Lintasan dari a ke j adalaha a, b, e, j
   Panjang lintasan dari a ke j adalah 3
3. **Saudara Kandung (Sibling)**
   f adalah saudara kandung e, tapi g bukan saudara kandung e, karena orang tua mereka berbeda
4. **Daun**
   Simpul yang berderajat nol (atau tidak mempunyai anak) disebut daun. Simpul h, i, j, f, c, l, m adalah daun.
5. **Simpul Dalam**
   Simpul yang mempunyai anak disebut simpul dalam. Simpul b, d, e, g, dan k adalah simpul dalam.

Pohon berakar yang setiap simpul cabangnya mempunyai paling banyak n buah anak disebut pohon n-ary.




**Gambar 1. Pohon Berakar**

Sebuah pohon dapat digunakan untuk memetakan pilihan keputusan. Pohon yang memetakan pilihan-pilihan keputusan tersebut dinamakan *decision tree* atau pohon keputusan. Contoh bentuk dari pohon keputusan dapat dilihat pada gambar 2.




**Gambar 2. Pohon Keputusan (*Decision Tree*)**

Pohon pada gambar 2 tersebut dapat dibaca seperti berikut ini :

- Jika orang tua datang, maka pergi ke Cinema
  atau

- Jika orang tua tidak datang dan hari cerah, maka bermain tenis
  atau
- Jika orang tua tidak datang dan hari berangin dan sedang kaya, maka belanja
  atau
- Jika orang tua tidak datang dan hari berangin dan sedang miskin, maka pergi ke Cinema
  atau
- Jika orang tua tidak datang dan hari hujan, maka tetap di rumah

Representasi pohon berakar pada pohon keputusan adalah sebagai berikut :

- Simpul dalam merepresentasikan tes atribut
- Daun merepresentasikan klasifikasi

## 2. LEARNING DECISION TREE MENGGUNAKAN ID3

### 2.1 Menspesifikasikan masalah

Permasalahan dalam learning decision tree dapat dijabarkan seperti berikut :

Kita mempunyai sekumpulan contoh yang dikategorikan menjadi beberapa kategori (keputusan-keputusan). Kita juga mempunyai sekumpulan atribut yang menjelaskan contoh-contoh yang ada, dan setiap atrubut memiliki nilai yang terhingga. Kita ingin menggunakan contoh-contoh yang ada untuk mempelajari struktur dari sebuah pohon keputusan yang dapat digunakan untuk menentukan kategori dari contoh yang belum ada (*unseen example*).

### 2.2 Basic Idea

Pada gambar 2, simpul ”parent visiting” diletakan paling atas (sebagai akar). Kita belum tahu mengapa, sebagaimana kita tidak tahu bagaimana proses pembuatan pohon keputusan tersebut. Namun, jika orang tua berkunjung, maka sudah pasti keputusannya adalah pergi ke cinema. Oleh karena itu, pilihan parent visiting bisa ditaruh paling atas tanpa memperdulikan kemungkinan yang lain.

Pemikiran diatas yang mendasari algoritma ID3.

### 2.3 Entropy

Membuat pohon keputusan adalah perkara memilih atribut mana yang harus diuji pada setiap simpul. Proses ini disebut information gain, yang berguna untuk menentukan atribut mana yang akan digunakan pada setiap simpul. Information Gain itu sendiri didapatkan dari perhitungan yang menggunakan satuan yang disebut entropy.

Rumus Entropy adalah :

$$Entropy(S) = -p_+ \, log_2(p_+) - p_- \, log_2(p_-)$$

$$Entropy(S) = \sum_{i=1}^{n} -p_i \, log_2(p_i)$$

### 2.4 Information Gain

Permasalahan sekarang adalah dalam menentukan atribut terbaik untuk simpul tertentu pada pohon. Pesamaan berikut ini menghitung sebuah nilai untuk atribut A, pada sebuah kumpulan contoh S. Nilai dari atribut A akan berkisar pada himpunan kemungkinan yang kita sebut Values(A). Untuk sebuah nilai v, yaitu sebuah nilai pada Values(A), kita dapat menuliskan $S_v$, yaitu himpunan contoh-contoh yang memiliki nilai v pada atribut A.

Information gain dari atribut A pada sebuah himpunan contoh S dapat dihitung dengan persamaan berikut :

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Sebagai contoh, misalnya kita memiliki sekumpulan contoh S = {s1,s2,s3,s4} yang dikategorikan menjadi positif dan negative, dimana s1 positif dan sisanya negative. Kita ingin menghitung information gain dari sebuah atribut A, dan A dapat memiliki nilai {v1,v2,v3}. Ditentukan bahwa :

s1 memiliki nilai v2 untuk A
s2 memiliki nilai v2 untuk A
s3 memiliki nilai v3 untuk A
s4 memiliki nilai v1 untuk A

Untuk mendapatkan information gain, pertama kita harus menghitung entropy dari S. Untuk menggunakan persamaan entropi pada persoalan ini, kita harus mengetahui jumlah positif dan negative pada S. Dari soal dapat diketahui bahwa positif = 1/4 dan negative = 3/4, sehingga kita bisa menghitung :

Entropy(S) = - (1/4)$\log_2$(1/4) – (3/4)$\log_2$(3/4) = - (1/4)(-2) – (3/4)(-0,415) = 0,5 + 0,311 = 0,811

Selanjutnya kita perlu menghitung Entropi($S_v$) untuk setiap nilai v = v1, v2, v3, v4. $S_v$ merupakan kumpulan dari contoh pada S yang memiliki nilai v pada atribut A, atau dapat ditulisakan sebagai berikut :

$S_{v1} = \{s_4\}$, $s_{v2}=\{s_1, s_2\}$, $s_{v3} = \{s_3\}$

Sekarang, kita dapat menyelesaikan persamaan-persamaan berikut :

(|Sv1|/|S|) * Entropy(Sv1) = (1/4) * (-(0/1)log2(0/1) - (1/1)log2(1/1)) = (1/4)(-0 -(1)log2(1)) = (1/4)(-0 -0) = 0

(|Sv2|/|S|) * Entropy(Sv2) = (2/4) * (-(1/2)log2(1/2) - (1/2)log2(1/2)) = (1/2) * (-(1/2)*(-1) - (1/2)*(-1)) = (1/2) * (1) = 1/2

(|Sv3|/|S|) * Entropy(Sv3) = (1/4) * (-(0/1)log2(0/1) - (1/1)log2(1/1)) = (1/4)(-0 -(1)log2(1)) = (1/4)(-0 -0) = 0

Sekarang kita bisa menambahkan ketiga nilai tersebut dan mendapatkan Entropi(S) untuk hasil akhir :

Gain(S,A) = 0.811 - (0 + 1/2 + 0) = 0.311

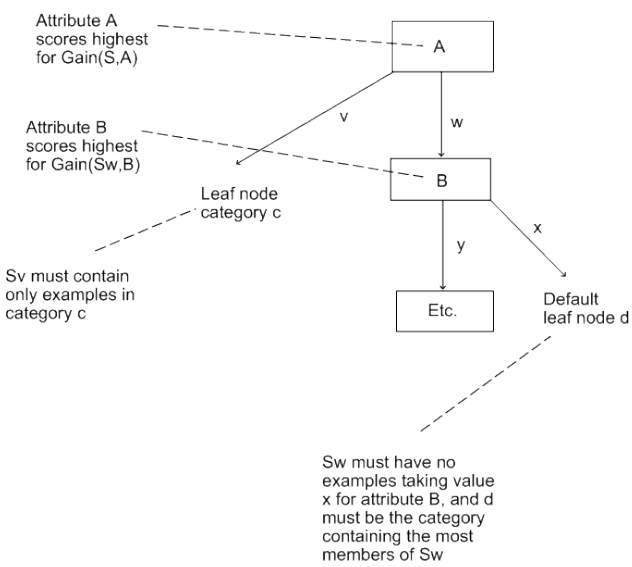
## 3. ALGORITMA ID3

Algoritma ID3 dapat dituliskan sebagai berikut :

Misal, diberikan sekumpulan contoh S, dikategorikan dalam sekumpulan kategori $c_i$, maka :

1. Tentukan simpul akar yang merupakan atribut A yang memiliki nilai information gain terbesar pada S
2. Untuk setiap nilai v yang mungkin dimiliki A, gambarkan cabang untuk simpul tersebut
3. Untuk setiap cabang dari A yang berkorespondensi dengan nilai v, hitung $S_v$.
   - Jika $S_v$ kosong, pilih kategori $c_{default}$ yang mempunyai paling banyak contoh pada S, dan letakan pada simpul daun yang mengakhiri cabang tersebut
   - Jika $S_v$ hanya mengandung contoh dari kategori c, maka letakan c sebagai simpul daun yang mengakhiri cabang tersebut
   - Selain itu, hilangkan A dari kumpulan atribut yang bisa diletakan pada simpul. Lalu, letakan simpul baru pada *decision tree*, dimana atribut baru yang diletakan adalah atribut yang memiliki information gain terbesar terhadap $S_v$. Ulangi langkah diatas dengan mengganti S dengan $S_v$

Algoritma diatas berhenti jika atribut sudah habis, atau *decision tree* sudah mengklasifikasi contoh dengan sempurna. Ilustrasi dari algoritma ID3 dapat dilihat pada gambar 3.




**Gambar 3. Algoritma ID3**

## 4. KESIMPULAN

Penggunaan pohon dalam bidang informatika sangat banyak. Salah satunya untuk membuat decision tree learning. Decision tree learning secara garis besar mirip dengan pohon n-ary dengan sedikit tambahan rumus-rumus perhitungan entropy dan information gain.

## REFERENSI


[1] Munir, Rinaldi. (2006). *Diktat Kuliah IF2153 Matematika Diskrit*. Program Studi Teknik Informatika, Institut Teknologi Bandung

[2] Russel, Norvig. (2003). *Artificial Inteligence – a Modern Approach 2nd Edition*. Pearson Education Inc, New Jersey

[3] http://www.doc.ic.ac.uk/~sgc/teaching/v231/lecture11.html waktu akses : 20 Desember 2009 20.00 WIB