

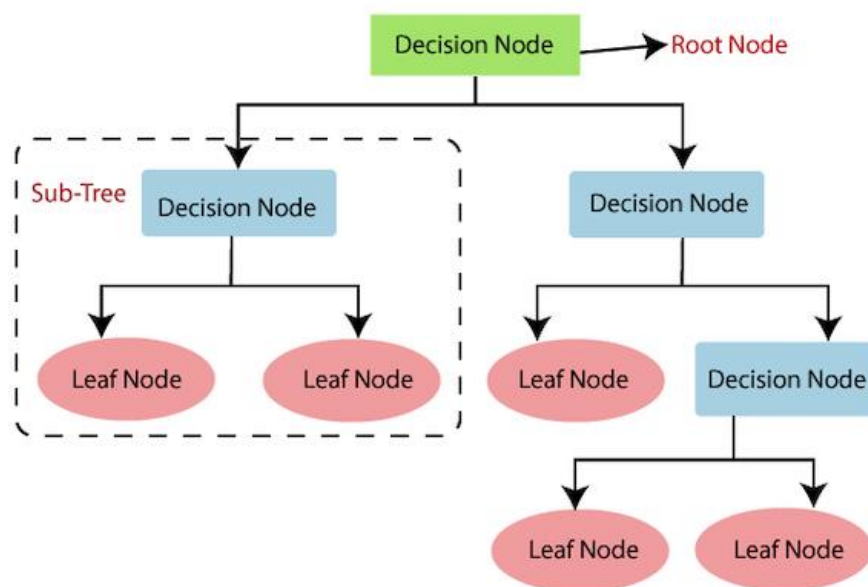
## BAB 7

### *Decision tree*

#### 7.1 Pengertian Decision tree

Decision tree merupakan salah satu model machine learning yang termasuk kedalam supervised learning. decision tree berbentuk seperti graph yang memodelkan keputusan. Digunakan untuk merepresentasikan keputusan dan pengambilan keputusan secara visual dan eksplisit. Decision tree disebut juga dengan istilah CART (Classification and Regression Trees) yang diperkenalkan oleh Leo Breiman untuk merujuk pada algoritma Decision Tree yang dapat digunakan untuk masalah klasifikasi atau regresi. Algoritma CART menyediakan dasar untuk algoritma penting seperti **bagged decision trees, random forest and boosted decision trees**.

Istilah penting dalam Decision tree



Gambar 7. 1 Decision tree

Model decision tree adalah binary tree dimana tree tersebut terdiri dari beberapa komponen yaitu :

- Setiap **Node** mewakili Fitur (Atribut/ variable  $x$ ) pada gambar 7.1 node disimbolkan dengan persegi Panjang



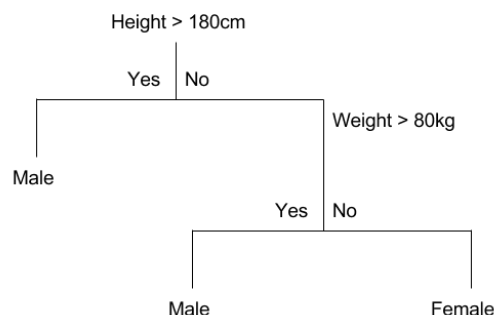
- b. Setiap **Branch** atau **Link** atau **Edge** mewakili Keputusan atau Aturan yang disimbolkan dengan panah
- c. Setiap **leaf** mewakili Hasil keputusan yang merupakan output variable digunakan sebagai hasil prediksi, disimbolkan dengan bentuk elips.

Selain komponen diatas terdapat beberapa istilah yang ada dalam decision tree

- a. **Splitting / Pemisahan**: Ini adalah proses membagi node menjadi dua atau lebih sub-node.
- b. **Decision Node**: Ketika sub-node terpecah menjadi sub-node lebih lanjut, maka itu disebut node keputusan.
- c. **Pruning / Pemangkasan**: Saat dilakukan penghapusan sub-node dari node keputusan, proses ini disebut pemangkasan.
- d. **Branch / Sub-Tree**: Sebuah sub-bagian dari seluruh pohon disebut cabang atau sub-pohon.
- e. **Parent and Child Node (Node Induk dan Anak)**: Node, yang dibagi menjadi beberapa sub-node disebut node induk dari sub-node sedangkan sub-node adalah anak dari node induk

Pada permasalahan klasifikasi leaf node dari decision tree mewakili kelas sedangkan pada permasalahan regresi nilai variabel respons untuk instance yang terdapat dalam leaf node dapat dirata-ratakan untuk menghasilkan estimasi untuk variabel respons. Cara untuk melakukan prediksi pada decision tree untuk data pengujian hanya perlu mengikuti edge sampai mencapai leaf node.

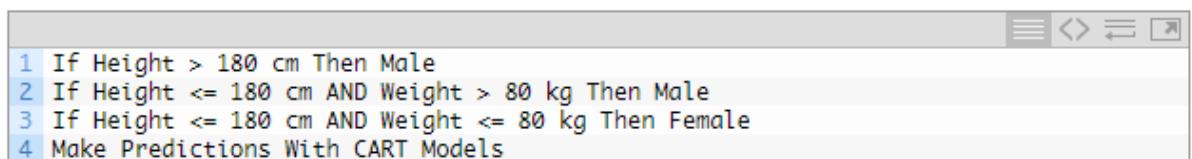
## 7.2 Contoh Decision tree



Gambar 7. 2 Contoh Decision tree



Gambar 7.2 merupakan contoh dari decision tree, contoh tersebut merupakan data fiktif yang bertujuan untuk memudahkan gambaran decision tree. Pada gambar 7.2 terdapat dua buah fitur/ input variable (x) yaitu height dalam satuan cm dan weight dalam satuan kg. contoh diatas merupakan tree yang digunakan untuk permasalahan klasifikasi yang memiliki dua buah keputusan/kelas yaitu kelas Male dan Kelas Female. Contoh tree pada gambar 7.2 juga dapat disimpan dalam bentuk file sebagai grafik atau seperangkat aturan seperti gambar 7.3



```
1 If Height > 180 cm Then Male
2 If Height <= 180 cm AND Weight > 80 kg Then Male
3 If Height <= 180 cm AND Weight <= 80 kg Then Female
4 Make Predictions With CART Models
```

Gambar 7. 3 Kumpulan Aturan Decision tree

### 7.3 Tahapan Algoritma Decision Tree

Pada Decision tree pembuatan tree merupakan Langkah awal yang harus dilaksanakan. Dalam pembuatan tree diperlukan perhitungan menggunakan Information Gain ataupun Gini Index untuk menentukan Root node/ node selanjutnya. Berikut merupakan tahapan pembuatan tree secara top-down:

- 1) Pada iterasi pertama dilakukan pemilihan Root Node didasarkan pada Gini Indeks terendah atau Information Gain tertinggi dari keseluruhan fitur (variable x) pada data training.
- 2) Kemudian memisahkan himpunan S untuk menghasilkan subset data
- 3) Selanjut proses akan diulang untuk mencari decision node yang akan menjadi cabang selanjutnya. Penentuan decision node juga menggunakan Gini Indeks terendah atau Information Gain tertinggi. Akan tetapi yang membedakan adalah fitur yang digunakan adalah fitur yang tidak pernah menggunakan sebelum nya (subset data)
- 4) Algoritma terus menerus berulang pada setiap subset data hingga menemukan leaf yang merupakan label kelas

### 7.4 Information Gain dengan menggunakan Entropy



Entropy digunakan untuk menghitung ketidakaturan dalam artian entropy akan menghitung homogenitas suatu fitur atribut (A) dari sebuah fitur yang ada di sample data (S). Entropy digunakan untuk mengukur jumlah ketidakpastian dalam variable. Berikut merupakan formula yang digunakan untuk mengukur entropy :

$$H(X) = - \sum_{i=1}^n P(x_i) \log_b P(x_i)$$

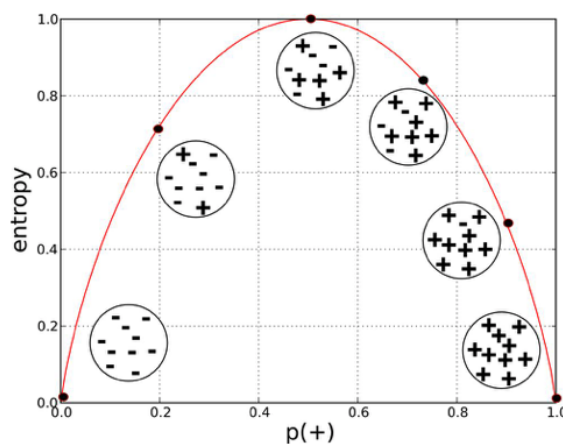
Dimana  $n$  merupakan berapa banyak outcome bisa berarti berapa banyak kelas

$P(x_i)$ , merupakan probabilitas dari outcome  $i$ .

$b$ , dapat bernilai 2, e, atau 10.

Karena log dari angka kurang dari 1 akan negatif, seluruh jumlah dinegasikan untuk menghasilkan nilai positif.

Penjumlahan dari total entropy bisa lebih dari 1.



Gambar 7. 4 Ilustrasi Entropy

Pada gambar 7.4 menunjukkan ilustrasi entropy untuk outcome/kelas yang memiliki dua buah label. Entropy(S) adalah jumlah bit yang diperkirakan dibutuhkan untuk dapat mengekstrak suatu kelas (+ atau -) dari sejumlah data acak pada ruang sample S. Entropy bisa dikatakan sebagai kebutuhan bit untuk menyatakan suatu kelas. Semakin kecil nilai Entropy maka semakin baik untuk digunakan dalam mengekstraksi suatu kelas.

Contoh : lemparan koin hanya akan memiliki dua outcome(kelas) yaitu kelas kepala dan kelas ekor. Probabilitas koin akan mendarat kepala adalah 0,5, dan probabilitas



koin akan mendarat di ekor adalah 0,5. Sehingga dari contoh tersebut didapatkan Entropi nya adalah :

$$\begin{aligned} H(X) &= - \sum_{i=1}^2 P(x_i) \log_b P(x_i) = -(P(x_1) \log_2 P(x_1) + P(x_2) \log_2 P(x_2)) \\ &= -(0.5 \cdot \log_2(0.5) + 0.5 \cdot \log_2(0.5)) = 1.0 \end{aligned}$$

Information Gain merupakan perbedaan antara entropi pada parent node,  $H(T)$ , dan rata-rata bobot dari entropi pada child node. Information Gain berdasarkan pada penurunan entropi setelah pemisahan kumpulan data pada atribut. Membangun decision tree adalah tentang bagaimana menentukan node berdasarkan information gain tertinggi (yaitu cabang yang paling homogen). Formula untuk menghitung information gain adalah sebagai berikut :

$$\begin{aligned} IG(T, a) &= H(T) - \sum_{v \in \text{vals}(a)} \frac{|\{x \in T | x_a \in v\}|}{|T|} H(\{x \in T | x_a \in v\}) \\ IG(T, a) &= H(T) - \sum_a \frac{|S_a|}{|T|} H(S_a) \end{aligned}$$

$T$  adalah kumpulan instance (banyaknya instance).  $a$  adalah fitur yang sedang diuji.

Pembuatan tree dengan menggunakan information gain

- a. Hitung entropy untuk sekumpulan dataset  $H(X)$
- b. Untuk setiap atribut / feature, lakukan
  1. Hitung entropy untuk setiap fitur  $x_i$
  2. Ambil Entropi Informasi Rata-rata dari Atribut / Fitur Saat Ini,  $x_i$
  3. Hitung information gain untuk Atribut / Fitur saat ini
- c. Pilih atribut / fitur yang memiliki information gain tertinggi
- d. Ulangi proses hingga mendapatkan tree yang di inginkan.

### Contoh Studi kasus pembuatan tree menggunakan information gain

Terdapat dataset seperti pada table 7.1. Data tersebut merupakan dataset untuk menentukan spesies Anjing / Kucing berdasarkan fitur play fetch, Is grumpy, dan Favourite food.



Tabel 7. 1 Dataset penentuan spesies

Training instance	Plays fetch	Is grumpy	Favorite food	species
1	Yes	No	Bacon	Dog
2	No	Yes	Dog food	Dog
3	No	Yes	Cat food	Cat
4	No	Yes	Bacon	Cat
5	No	No	Cat food	Cat
6	No	Yes	Bacon	Cat
7	No	Yes	Cat food	Cat
8	No	No	Dog food	Dog
9	No	Yes	Cat food	Cat
10	Yes	No	Dog food	Dog
11	Yes	No	Bacon	Dog
12	No	No	Cat food	Cat
13	Yes	Yes	Cat food	Cat
14	Yes	Yes	Bacon	Dog

Tahapan pembuatan Tree

(1) Iterasi Pertama

a. Hitung Entropy untuk dataset

$$H(X) = -\left(\frac{8}{14} \log_2 \left(\frac{8}{14}\right) + \frac{6}{14} \log_2 \left(\frac{6}{14}\right)\right) = 0.9852$$

b. Hitung entropy dan information gain untuk setiap atribut/ fitur, sehingga nantinya akan didapatkan entropy untuk Attribute(Plays fetch)  $x_1$ , entropy untuk Attribute(Is grumpy)  $x_2$ , entropy untuk Attribute(Favorite food)  $x_3$

Pada atribut play fetch

$$\begin{aligned} H(x_{1,yes}): [1c, 4d] \rightarrow H(x_{1,yes}) &= -\left(\frac{1}{5} \log_2 \left(\frac{1}{5}\right) + \frac{4}{5} \log_2 \left(\frac{4}{5}\right)\right) \\ &= -(-0.4644 - 0.2575) = 0.7219 \end{aligned}$$

$$\begin{aligned} H(x_{1,no}): [7c, 2d] \rightarrow H(x_{1,no}) &= -\left(\frac{7}{9} \log_2 \left(\frac{7}{9}\right) + \frac{2}{9} \log_2 \left(\frac{2}{9}\right)\right) \\ &= -(-0.2819 - 0.4822) = 0.7641 \end{aligned}$$

$$\begin{aligned} IG(X, Plays fetch) &= H(X) - \frac{5}{14}(0.7219) - \frac{9}{14}(0.7641) \\ &= 0.9852 - 0.2578 - 0.4912 = 0.2362 \end{aligned}$$



Pada atribut Is grumpy

$$\begin{aligned}
 H(x_{2,yes}): [6c, 2d] &\rightarrow H(x_{2,yes}) = -\left(\frac{6}{8}\log_2\left(\frac{6}{8}\right) + \frac{2}{8}\log_2\left(\frac{2}{8}\right)\right) \\
 &= -(-0.3113 - 0.5) = 0.8113 \\
 H(x_{2,no}): [2c, 4d] &\rightarrow H(x_{2,no}) = -\left(\frac{2}{6}\log_2\left(\frac{2}{6}\right) + \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right) \\
 &= -(-0.5283 - 0.3899) = 0.9182 \\
 IG(X, Is grumpy) &= H(X) - \frac{8}{14}(0.8113) - \frac{6}{14}(0.9182) \\
 &= 0.9852 - 0.4636 - 0.3935 = 0.1281
 \end{aligned}$$

Pada atribut Favourite Food

untuk hal ini atribut dipisahkan menjadi 3 atribut nilai biner yang berbeda yaitu bacon, cat food, dan dog food

- **Sub-Attribute[bacon]**

$$\begin{aligned}
 H(x_{3,bacon}): [5+, 9-] \\
 H(x_{3,+bacon}): [2c, 3d] &\rightarrow H(x_{3,+bacon}) = -\left(\frac{2}{5}\log_2\left(\frac{2}{5}\right) + \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right) \\
 &= -(-0.5287 - 0.4421) = 0.9708. \\
 H(x_{3,-bacon}): [6c, 3d] &\rightarrow H(x_{3,-bacon}) = -\left(\frac{6}{9}\log_2\left(\frac{6}{9}\right) + \frac{3}{9}\log_2\left(\frac{3}{9}\right)\right) \\
 &= -(-0.39 - 0.5283) = 0.9183. \\
 IG(X, bacon) &= H(X) - \frac{5}{14}(0.9708) - \frac{9}{14}(0.9183) \\
 &= 0.9852 - 0.3467 - 0.5904 = 0.0481
 \end{aligned}$$

- **Sub-Attribute(dog food):**

$$\begin{aligned}
 H(x_{3,dog food}): [3+, 11-] \\
 H(x_{3,+dog food}): [0c, 3d] &\rightarrow H(x_{3,+dog food}) = -\left(\frac{0}{3}\log_2\left(\frac{0}{3}\right) + \frac{3}{3}\log_2\left(\frac{3}{3}\right)\right) \\
 &= -(0 - 0) = 0. \\
 H(x_{3,-dog food}): [8c, 3d] &\rightarrow H(x_{3,-dog food}) = -\left(\frac{8}{11}\log_2\left(\frac{8}{11}\right) + \frac{3}{11}\log_2\left(\frac{3}{11}\right)\right) \\
 &= -(0.3343 - 0.5113) = 0.8456.
 \end{aligned}$$



$$IG(X, dog\ food) = H(X) - \frac{3}{14}(0) - \frac{11}{14}(0.8456) = 0.9852 - 0 - 0.6644 \\ = 0.3210$$

• **Sub-Attribute(cat food):**

$$H(x_{3,cat\ food}): [6+, 8-]$$

$$H(x_{3,+cat\ food}): [6c, 0d] \rightarrow H(x_{3,+cat\ food}) = -\left(\frac{6}{6}\log_2\left(\frac{6}{6}\right) + \frac{0}{6}\log_2\left(\frac{0}{6}\right)\right) \\ = -(0 - 0) = 0.$$

$$H(x_{3,-cat\ food}): [2c, 6d] \rightarrow H(x_{3,-cat\ food}) = -\left(\frac{2}{8}\log_2\left(\frac{2}{8}\right) + \frac{6}{8}\log_2\left(\frac{6}{8}\right)\right) \\ = -(0.5 - 0.3113) = 0.8113.$$

$$IG(X, cat\ food) = H(X) - \frac{6}{14}(0) - \frac{8}{14}(0.8113) = 0.9852 - 0 - 0.4636 \\ = 0.5216$$

c. Pilih atribut / fitur yang memiliki information gain tertinggi

$$IG(X, Plays\ fetch) = 0.2362$$

$$IG(X, Is\ grumpy) = 0.1281$$

$$IG(X, bacon) = 0.0481$$

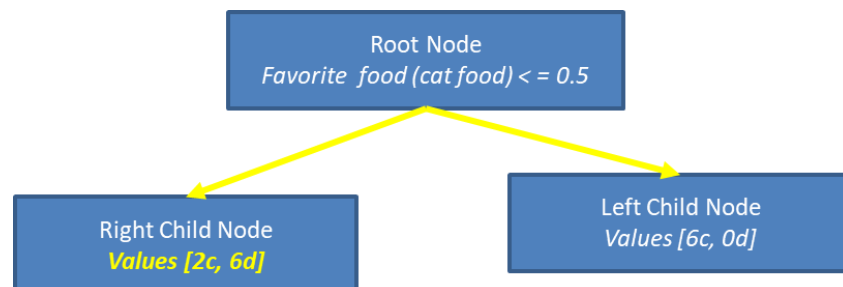
$$IG(X, dog\ food) = 0.3210$$

$$IG(X, cat\ food) = \mathbf{0.5216}$$

Information Gain tertinggi dimiliki oleh sub atribut *Favorite food (cat food)*.

Iterasi pertama akan dimulai **dengan Root Node** = *Favorite food (cat food)*.

Sehingga tree yang terbentuk pada iterasi pertama adalah



(2) Iterasi Kedua

Setelah ditemukan root node pada iterasi pertama maka pada iterasi kedua pencarian node dilakukan setelah memisahkan himpunan S untuk menghasilkan subset data. Subset data yang terbentuk Ketika favorite food(cat food) bernilai false ditunjukkan





pada table 7.2, sedangkan subset data ketika favorite food(cat food) bernilai true ditunjukkan oleh table 7.3

Tabel 7. 2 Subset data pada iterasi 2 favorite food( bukan cat food)

Instance	No Cat Food	Plays fetch	Is grumpy	Species
1	Bacon	Yes	No	Dog
2	Dog food	No	Yes	Dog
4	Bacon	No	Yes	Cat
6	Bacon	No	Yes	Cat
8	Dog food	No	No	Dog
10	Dog food	Yes	No	Dog
11	Bacon	Yes	No	Dog
14	Bacon	Yes	Yes	Dog

- a. Hitung Entropy untuk subdataset

$$H(X) = -\left(\frac{6}{8} \log_2 \left(\frac{6}{8}\right) + \frac{2}{8} \log_2 \left(\frac{2}{8}\right)\right) = 0,811278$$

- b. Hitung entropy dan information gain untuk setiap atribut/ fitur, sehingga nantinya akan didapatkan entropy untuk Attribute(Plays fetch) x1, entropy untuk Attribute(Is grumpy) x2, entropy untuk Attribute(Favorite food) x3

Pada atribut play fetch

$$(x_{1,yes}): [0c, 4d] \rightarrow H(x_{1,yes}) = -\left(\frac{0}{4} \log_2 \left(\frac{0}{4}\right) + \frac{4}{4} \log_2 \left(\frac{4}{4}\right)\right) = 0$$

$$H(x_{1,no}): [2c, 2d] \rightarrow H(x_{1,no}) = -\left(\frac{2}{4} \log_2 \left(\frac{2}{4}\right) + \frac{2}{4} \log_2 \left(\frac{2}{4}\right)\right) = 1$$

$$IG(X, Plays fetch) = H(X) - \frac{4}{8}(0) - \frac{4}{8}(1) = 0,811278 - 0 - 0,5 = 0,311278$$

Pada atribut Is grumpy

$$H(x_{2,yes}): [2c, 2d] \rightarrow H(x_{2,yes}) = -\left(\frac{2}{4} \log_2 \left(\frac{2}{4}\right) + \frac{2}{4} \log_2 \left(\frac{2}{4}\right)\right) = 1$$

$$H(x_{2,no}): [0c, 4d] \rightarrow H(x_{2,no}) = -\left(\frac{0}{4} \log_2 \left(\frac{0}{4}\right) + \frac{4}{4} \log_2 \left(\frac{4}{4}\right)\right) = 0$$

$$IG(X, Is grumpy) = H(X) - \frac{4}{8}(1) - \frac{4}{8}(0) = 0,811278 - 0 - 0,5 = 0,311278$$

Pada atribut Favorite Food



$$(x_{3,bacon}): [2c, 3d] \rightarrow H(x_{3,bacon}) = -\left(\frac{2}{5} \log_2 \left(\frac{2}{5}\right) + \frac{3}{5} \log_2 \left(\frac{3}{5}\right)\right) \\ = 0,970950594$$

$$H(x_{3,dogfood}): [0c, 3d] \rightarrow H(x_{3,bacon}) = -\left(\frac{0}{3} \log_2 \left(\frac{0}{3}\right) + \frac{3}{3} \log_2 \left(\frac{3}{3}\right)\right) = 0$$

$$IG(X, favoritefood) = H(X) - \frac{5}{8}(0,970950594) - \frac{3}{8}(0) = 0,811278 - 0 - 0 \\ = 0,204433878$$

- c. Pilih atribut / fitur yang memiliki information gain tertinggi

$$IG(X, Plays fetch) = 0,311278$$

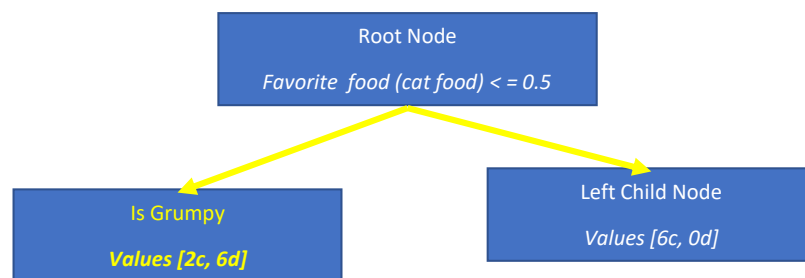
$$IG(X, Is grumpy) = 0,311278$$

$$IG(X, favoritefood) = 0,204433878$$

Information Gain tertinggi dimiliki oleh atribut *plays fetch* dan *is grumpy*. Pilih salah satu untuk menjadi node misalkan *is grumpy*

Iterasi kedua ketika favorite food(cat food) bernilai false **Node** = *is grumpy*.

Sehingga tree yang terbentuk pada iterasi kedua ketika favorite food(cat food) bernilai false adalah :

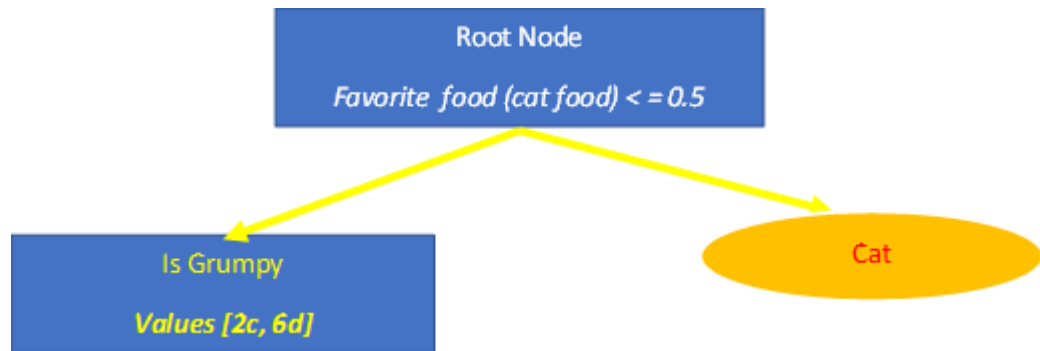


Tabel 7. 3 Subset data pada iterasi 2 favorite food(cat food)

instance	play fetch	Is grumpy	favorite food	spesies
3	no	yes	cat food	cat
5	no	no	cat food	cat
7	no	yes	cat food	cat
9	no	yes	cat food	cat
12	no	no	cat food	cat
13	yes	yes	cat food	cat



Ketika favorite food(cat food) bernilai true dapat dilihat pada tabel 7.3 bahwa keseluruhan outcome/kelas pada kolom spesies adalah cat. Yang artinya Ketika terdapat instance yang memiliki favorite food bernilai **cat** dapat dipastikan bahwa spesiesnya adalah cat food. Sehingga tree yang terbentuk adalah :



### (3) Iterasi Ketiga

Setelah ditemukan rood node pada iterasi pertama dan sub node pada serta leaf node pada iterasi 2 maka pada iterasi ketiga pencarian node dilakukan setelah memisahkan himpunan S untuk menghasilkan subset data. Subset data yang terbentuk Ketika favorite food(cat food) bernilai false, is grumpy yes ditunjukkan pada table 7.4, sedangkan subset data ketika favorite food(cat food) bernilai false, is grumpy no ditunjukkan oleh table 7.5

Tabel 7. 4 Subset data food(cat food) bernilai false, is grumpy yes

instance	play fetch	Is grumpy	favorite food	spesies
2	no	yes	dog food	dog
4	no	yes	bacon	cat
6	no	yes	bacon	cat
14	yes	yes	bacon	dog

Lakukan perhitungan seperti iterasi 1 dan 2 untuk mendapatkan sub node selanjutnya pada data subset table 7.4. berdasarkan table 7.4 didapatkan IG tertinggi adalah pada:

$$IG(X, Plays\ fetch) = 0,311278$$

$$IG(X, favoritefood) = 0,311278$$

Information Gain yang dimiliki oleh atribut *plays fetch* dan *favoritefood* bernilai sama.

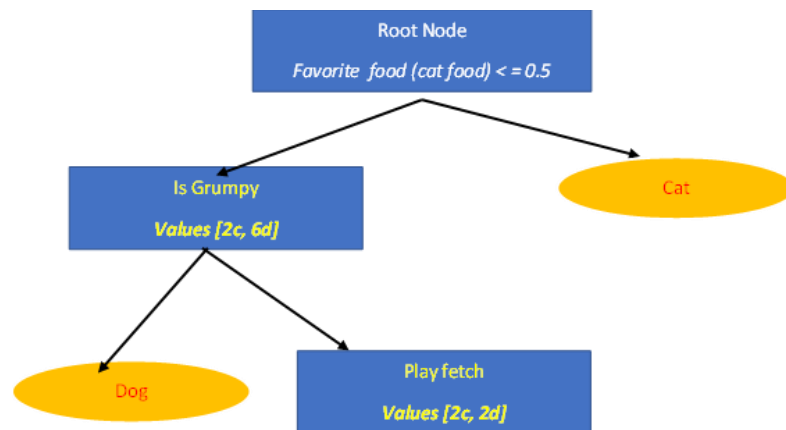
Pilih salah satu untuk menjadi node misalkan *play fetch*



Tabel 7. 5 Subset data food(cat food) bernilai false, is grumpy yes

instance	play fetch	Is grumpy	favorite food	spesies
1	yes	no	bacon	dog
8	no	no	dog food	dog
10	yes	no	dog food	dog
11	yes	no	bacon	dog

Pada tabel 7.5 bahwa keseluruhan outcome/kelas pada kolom spesies adalah cat. Yang artinya Ketika terdapat instance yang memiliki favorite food bernilai *data food (cat food)* bernilai *false* dan *is grumpy yes* dapat dipastikan bahwa spesiesnya adalah Dog Sehingga tree yang terbentuk dari iterasi 3 adalah :



#### (4) Iterasi Keempat

Lakukan Langkah seperti iterasi 1,2,3 untuk iterasi 4 dengan menggunakan subset data pada tabel 7.6 dan tabel 7.7

Tabel 7. 6 subset data iterasi 4 palyfetch = no, is grumpy =yes

instance	play fetch	Is grumpy	favorite food	spesies
2	no	yes	dog food	dog
4	no	yes	bacon	cat
6	no	yes	bacon	cat

Tabel 7. 7 subset data iterasi 4 palyfetch = yes, is grumpy =yes

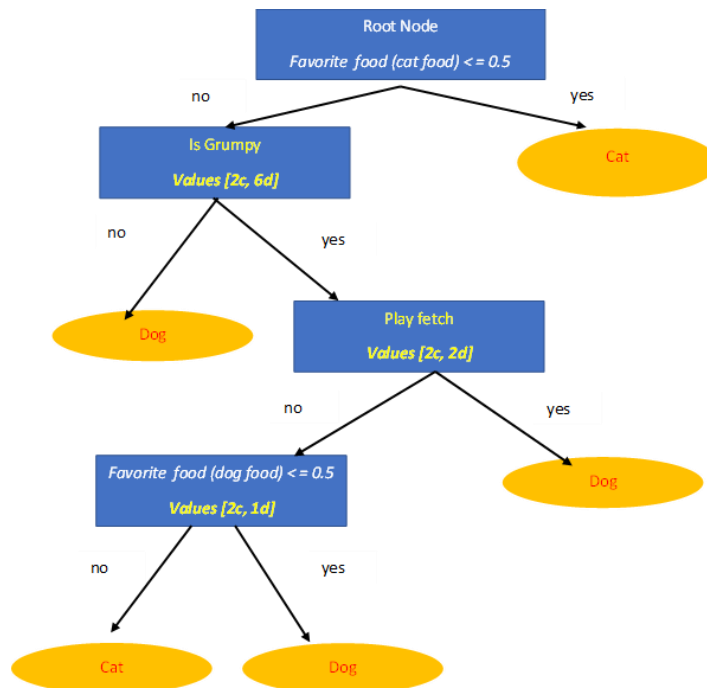
instance	play fetch	Is grumpy	favorite food	spesies
14	yes	yes	bacon	dog

#### (5) Iterasi Kelima



instance	play fetch	Is grumpy	favorite food	spesies
2	no	yes	dog food	dog
4	no	yes	bacon	cat
6	no	yes	bacon	cat

Hasil Akhir Tree



Sehingga ketika ada data testing yang perlu dilakukan adalah mengikuti tree yang sudah terbentuk, missal data testing tersebut.

play fetch	Is grumpy	favorite food
yes	yes	bacon

Maka setelah mengikuti tree hasil keputusan yang didapat data training tersebut merupakan spesies "Dog".

## 7.5 Gini Index

Heuristik umum untuk mempelajari decision tree dan Mengukur proporsi kelas dalam satu set. Gini index dihitung dengan mengurangi jumlah probabilitas kuadrat setiap kelas. Formula gini index :



$$Gini(t) = 1 - \sum_{i=1}^j P(i|t)^2$$

Dimana  $j$  adalah jumlah kelas.

$t$  adalah bagian dari instance untuk node.

$P(i | t)$  adalah probabilitas memilih elemen kelas  $i$  dari subset node.

Pada Gini index penentuan node berdasarkan nilai gini index paling kecil, CART memilih node tersebut sebagai node Keputusan, dan node tersebut merupakan fitur tidak pernah digunakan pada node sebelumnya. Proses berulang ini berhenti sampai pohon Keputusan mendapatkan leaf node

### Contoh Studi kasus pembuatan tree menggunakan Gini index

Terdapat dataset seperti pada table 7.8. Data tersebut merupakan dataset untuk menentukan spesies Mammal / Reptile berdasarkan Toothed, Hair, Breathes, Legs.

Tabel 7. 8 Dataset untuk menentukan Spesies

Toothed	Hair	Breathes	Legs	Species
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Not Hair	Breathes	Not Legs	Reptile
Not Toothed	Hair	Breathes	Legs	Mammal
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Not Hair	Not Breathes	Not Legs	Reptile
Toothed	Not Hair	Breathes	Not Legs	Reptile
Toothed	Not Hair	Breathes	Legs	Mammal
Not Toothed	Not Hair	Breathes	Legs	Reptile

### LANGKAH PERTAMA:

Menghitung Gini Indeks untuk semua fitur.

Tabel 7.9 Penghitungan Gini Index Toothed langkah pertama

Toothed			
	Species		Total
	Mammal	Reptile	
Toothed	5	3	8
Not Toothed	1	1	2
TOTAL			10

$$Gini(\text{Toothed}=\text{Toothed}) = 1 - (5/8)^2 - (3/8)^2 = 1 - 0.39 - 0.14 = 0.46875$$



$$\text{Gini}(\text{Toothed}=\text{Not Toothed}) = 1 - (1/2)^2 - (1/2)^2 = 1 - 0.25 - 0.25 = 0.5$$

$$\text{Weighted Gini Index}(\text{Toothed}) = (8/10) \times 0.46875 = 0.375$$

$$\text{Weighted Gini Index}(\text{Not Toothed}) = (2/10) \times 0.5 = 0.1$$

$$\text{Gini}(\text{Toothed}) = 0.375 + 0.1 = 0.475$$

Tabel 7.10 Hasil Penghitungan Gini Index Toothed langkah pertama

Features	Gini Index	Weighted Gini Index
Toothed	0.46875	0.375
Not Toothed	0.5	0.1
<b>Gini Index</b>		<b>0.475</b>

Tabel 7.11 Penghitungan Gini Index Hair langkah pertama

Hair			
	Species		Total
	Mammal	Reptile	
Hair	5	0	5
Not Hair	1	4	5
<b>TOTAL</b>			<b>10</b>

Tabel 7.12 Hasil Penghitungan Gini Index Hair langkah pertama

Features	Gini Index	Weighted Gini Index
Hair	0	0
Not Hair	0.32	0.16
<b>Gini Index</b>		<b>0.16</b>

Tabel 7.13 Penghitungan Gini Index Breathes langkah pertama

Breathes			
	Species		Total
	Mammal	Reptile	
Breathes	6	3	9
Not Breathes	0	1	1
<b>TOTAL</b>			<b>10</b>

Tabel 7.14 Hasil Penghitungan Gini Index Breathes langkah pertama

Features	Gini Index	Weighted Gini Index
Breathes	0.4444444	0.4
Not Breathes	0	0
<b>Gini Index</b>		<b>0.4</b>



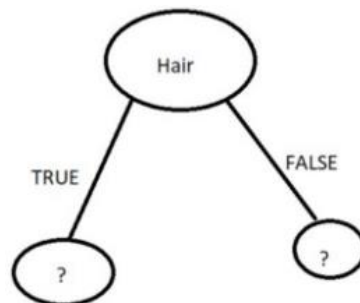
Tabel 7.15 Penghitungan Gini Index Legs langkah pertama

Legs			
	Species		Total
	Mammal	Reptile	
Legs	6	1	7
Not Legs	0	3	3
TOTAL			10

Tabel 7.16 Hasil Penghitungan Gini Index Legs langkah pertama

Features	Gini Index	Weighted Gini Index
Legs	0.244898	0.171428571
Not Legs	0	0
Gini Index		0.171428571

Hair dipilih sebagai Root node karena memiliki Indeks Gini paling sedikit. Untuk di mana kita berdiri, perlu divisualisasikan untuk melihat Pohon itu.



## LANGKAH KE DUA

Memisahkan tree / pohon berdasarkan Hair sama dengan true atau false dengan menghitung Indeks Gini.

(1) Pertama, mempertimbangkan skenario Hair = true

Tabel 7.17 Tabel skenario Hair = True

Toothed	Hair	Breathes	Legs	Species
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Hair	Breathes	Legs	Mammal
Not Toothed	Hair	Breathes	Legs	Mammal
Toothed	Hair	Breathes	Legs	Mammal
Toothed	Hair	Breathes	Legs	Mammal

Tabel 7.18 Penghitungan Gini Index Toothed langkah kedua

<b>Toothed</b>
----------------





	Species		Total
	Mammal	Reptile	
Toothed	4	0	4
Not Toothed	1	0	1
TOTAL			5

Tabel 7.19 Hasil Penghitungan Gini Index Toothed langkah kedua

Features	Gini Index	Weighted Gini Index
Toothed	0	0
Not Toothed	0	0
Gini Index		0

Tabel 7.20 Penghitungan Gini Index Breathes langkah kedua

Breathes			
	Species		Total
	Mammal	Reptile	
Breathes	5	0	5
Not Breathes	0	0	0
TOTAL			5

Tabel 7.21 Hasil Penghitungan Gini Index Breathes langkah kedua

Features	Gini Index	Weighted Gini Index
Breathes	0	0
Not Breathes	0	0
Gini Index		0

Tabel 7.22 Penghitungan Gini Index Legs langkah kedua

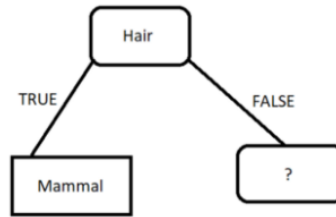
Legs			
	Species		Total
	Mammal	Reptile	
Legs	5	0	5
Not Legs	0	0	0
TOTAL			5

Tabel 7.23 Hasil Penghitungan Gini Index Legs langkah kedua

Features	Gini Index	Weighted Gini Index
Legs	0	0
Not Legs	0	0
Gini Index		0



Dari perhitungan di atas, semua Indeks Gini:  $G.I(\text{Toothed}) = G.I(\text{Hair}) = G.I(\text{Breathes}) = G.I(\text{Legs}) = 0$  jadi kita menyimpulkan bahwa itu adalah Leaf Node (simpul Daun) dan tidak dapat dipisahkan lebih jauh.



### LANGKAH KETIGA

Mempertimbangkan skenario Hair = false

Tabel 7.24 Tabel skenario Hair = false

Toothed	Hair	Breathes	Legs	Species
Toothed	Not Hair	Breathes	Not Legs	Reptile
Toothed	Not Hair	Not Breathes	Not Legs	Reptile
Toothed	Not Hair	Breathes	Not Legs	Reptile
Toothed	Not Hair	Breathes	Legs	Mammal
Not Toothed	Not Hair	Breathes	Legs	Reptile

Tabel 7.25 Penghitungan Gini Index Toothed langkah ketiga

Toothed			
	Species		Total
	Mammal	Reptile	
Toothed	1	3	4
Not Toothed	0	1	1
TOTAL			5

$$Gini(\text{Toothed}=\text{Toothed}) = 1 - (1/4)^2 - (3/4)^2 = 1 - 0.0625 - 0.5625 = 0.375$$

$$Gini(\text{Toothed}=\text{Not Toothed}) = 1 - (0/1)^2 - (1/1)^2 = 1 - 0 - 1 = 0$$

$$\text{Weighted Gini Index}(\text{Toothed}) = (4/5) \times 0.375 = 0.3$$

$$\text{Weighted Gini Index}(\text{Not Toothed}) = (1/5) \times 0 = 0$$

$$Gini(\text{Toothed}) = 0.3 + 0 = 0$$

Tabel 7.26 Hasil Penghitungan Gini Index Toothed langkah ketiga

Features	Gini Index	Weighted Gini Index
Toothed	0.375	0.3
Not Toothed	0	0



<b>Gini Index</b>	<b>0.3</b>
-------------------	------------

Tabel 7.27 Penghitungan Gini Index Breathes langkah ketiga

<b>Breathes</b>			
	<b>Species</b>		<b>Total</b>
	<b>Mammal</b>	<b>Reptile</b>	
Breathes	1	3	4
Not Breathes	0	1	1
<b>TOTAL</b>			<b>5</b>

Tabel 7.28 Hasil Penghitungan Gini Index Breathes langkah ketiga

<b>Features</b>	<b>Gini Index</b>	<b>Weighted Gini Index</b>
Breathes	0.375	0.3
Not Breathes	0	0
<b>Gini Index</b>		<b>0.3</b>

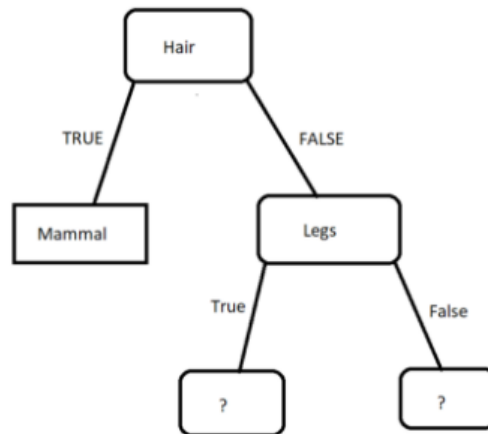
Tabel 7.29 Penghitungan Gini Index Legs langkah ketiga

<b>Legs</b>			
	<b>Species</b>		<b>Total</b>
	<b>Mammal</b>	<b>Reptile</b>	
Legs	1	1	2
Not Legs	0	3	3
<b>TOTAL</b>			<b>5</b>

Tabel 7.30 Hasil Penghitungan Gini Index Legs langkah ketiga

<b>Features</b>	<b>Gini Index</b>	<b>Weighted Gini Index</b>
Legs	0.5	0.2
Not Legs	0	0
<b>Gini Index</b>		<b>0.2</b>

Namun, ketika skenario Hair = false, Legs mempunyai nilai Gini Index terkecil sehingga di pilih sebagai Root Node dan Tree sekarang terlihat seperti gambar di bawah.



## LANGKAH KE EMPAT

Mempertimbangkan Skenario Ketika Hair = false dan Legs = true. Menghitung indeks Gini hanya untuk Toothed and Breathes.

Tabel 7.31 Tabel skenario Hair = false dan Legs = True

Toothed	Hair	Breathes	Legs	Species
Toothed	Not Hair	Breathes	Legs	Mammal
Not Toothed	Not Hair	Breathes	Legs	Reptile

Tabel 7.32 Penghitungan Gini Index Toothed langkah keempat

Toothed			
	Species		Total
	Mammal	Reptile	
Toothed	1	0	1
Not Toothed	0	1	1
TOTAL			2

Tabel 7.33 Hasil Penghitungan Gini Index Toothed langkah keempat

Features	Gini Index	Weighted Gini Index
Toothed	0	0
Not Toothed	0	0
Gini Index		0

Tabel 7.34 Penghitungan Gini Index Breathes langkah keempat

Breathes			
	Species		Total
	Mammal	Reptile	



Breathes	1	1	2
Not Breathes	0	0	0
<b>TOTAL</b>			<b>2</b>

Tabel 7.35 Hasil Penghitungan Gini Index Breathes langkah keempat

Features	Gini Index	Weighted Gini Index
Breathes	0.5	0.5
Not Breathes	0	0
<b>Gini Index</b>		<b>0.5</b>

Berdasarkan nilai Gini Index terkecil maka kita memilih Toothed sebagai Root node.

### LANGKAH KE LIMA

Mempertimbangkan skenario ketika Hair = false and Legs = false.

Tabel 7.36 Tabel skenario Hair = false dan Legs = false

Toothed	Hair	Breathes	Legs	Species
Toothed	Not Hair	Breathes	Not Legs	Reptile
Toothed	Not Hair	Not Breathes	Not Legs	Reptile
Toothed	Not Hair	Breathes	Not Legs	Reptile

Tabel 7.38 Hasil Penghitungan Gini Index Toothed langkah kelima

Features	Gini Index	Weighted Gini Index
Toothed	0	0
Not Toothed	0	0
<b>Gini Index</b>		<b>0</b>

Tabel 7.39 Penghitungan Gini Index Breathes langkah kelima

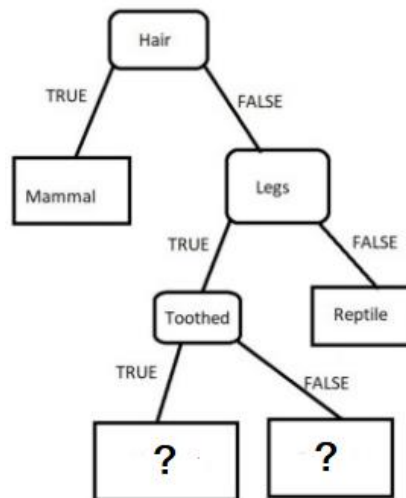
Breathes			
	Species		Total
	Mammal	Reptile	
Breathes	0	2	2
Not Breathes	0	1	1
<b>TOTAL</b>			<b>3</b>

Tabel 7.41 Hasil Penghitungan Gini Index Breathes langkah kelima

Features	Gini Index	Weighted Gini Index
Breathes	0	0
Not Breathes	0	0
<b>Gini Index</b>		<b>0</b>



Hasil perhitungan menunjukkan bahwa fitur Tothead dan Breathes adalah Leaf node karena masing-masing mempunyai nilai Gini = 0. Update tree terbaru adalah terlihat pada gambar di bawah.



## LANGKAH KE ENAM

- (1) Mempertimbangkan Skenario Ketika Hair = False, Legs=True & Toothed = True. Menghitung indeks Gini hanya untuk Breathes.

Tabel 7.42 Tabel skenario Hair = False Legs =True Toothed = true

Toothed	Hair	Breathes	Legs	Species
Toothed	Not Hair	Breathes	Legs	Mammal

Tabel 7.43 Penghitungan Gini Index Breathes langkah keenam

Breathes			
	Species		Total
	Mammal	Reptile	
Breathes	1	0	1
Not Breathes	0	0	0
TOTAL			1

Tabel 7.44 Hasil Penghitungan Gini Index Breathes langkah keenam

Features	Gini Index	Weighted Gini Index
Breathes	0	0
Not Breathes	0	0
Gini Index		0



- (2) Mempertimbangkan Skenario Ketika Hair = False, Legs=True & Toothed = False.  
Menghitung indeks Gini hanya untuk Breathes.

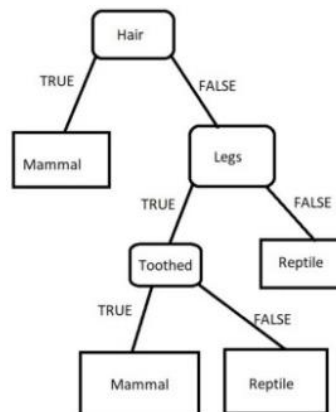
Tabel 7.45 Penghitungan Gini Index Breathes langkah keenam

Breathes			
	Species		Total
	Mammal	Reptile	
Breathes	0	1	1
Not Breathes	0	0	0
<b>TOTAL</b>			<b>1</b>

Tabel 7.46 Hasil Penghitungan Gini Index Breathes langkah keenam

Features	Gini Index	Weighted Gini Index
Breathes	0	0
Not Breathes	0	0
<b>Gini Index</b>	<b>0</b>	

Dalam kedua situasi di atas Gini Index = 0 berarti bahwa keduanya adalah Leaf node, sekarang kita dapat membuat Tree terakhir kita, sebagai berikut:



## 7.6 Kode program Python

```
[ ] 1 import pandas as pd
    2 import io
    3
    4 from google.colab import files
    5 upload=files.upload()

Choose Files diabetes.csv
• diabetes.csv(application/vnd.ms-excel) - 23873 bytes, last modified: 4/9/2021 - 100% done
Saving diabetes.csv to diabetes.csv
```



```
1 import pandas as pd
2 col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
3 # load dataset
4
5 pima = pd.read_csv(io.BytesIO(upload['diabetes.csv']), delimiter=',', header=None, names=col_names, skiprows = 1)
6 print(df.head())
7
```

	Pregnancies	Glucose	BloodPressure	...	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	...	0.627	50	1
1	1	85	66	...	0.351	31	0
2	8	183	64	...	0.672	32	1
3	1	89	66	...	0.167	21	0
4	0	137	40	...	2.288	33	1

[5 rows x 9 columns]

```
1 import pandas as pd
2 from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
3 from sklearn.model_selection import train_test_split # Import train_test_split function
4 from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
5
6 X = pima[feature_cols] # Features
7 y = pima.label # Target variable
8
9 # Split dataset into training set and test set
10 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=1)
11 clf = DecisionTreeClassifier()
12 clf = clf.fit(X_train,y_train)
13 #Predict the response for test dataset
14 y_pred = clf.predict(X_test)
15
16 # Model Accuracy, how often is the classifier correct?
17 print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

### Visualisasi decision tree

```
1 from sklearn.externals.six import StringIO
2 from IPython.display import Image
3 from sklearn.tree import export_graphviz
4 import pydotplus
5 dot_data = StringIO()
6 export_graphviz(clf, out_file=dot_data,
7                 filled=True, rounded=True,
8                 special_characters=True, feature_names = feature_cols, class_names=['0','1'])
9 graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
10 graph.write_png('diabetes.png')
11 Image(graph.create_png())
```

