

# Classification & Regression Trees

**Aruni Yasmin Azizah\*, Adila Alfa Krisnadhi, Siti Aminah,  
Dina Chahyati, Fariz Darari**

**CSGE603130: Kecerdasan Artifisial dan Sains Data Dasar  
Gasal 2022/2023**

# Outline

1. Learning Methods
2. Classification & Regression Tree
3. Contoh: Classification Tree
4. Contoh: Regression Tree
5. Issues, Pros and Cons of CART

# Learning Methods

Sumber:

- Slides Materi Sistem Cerdas, “Machine Learning: Decision Trees”, Semester Genap 2020/2021
- Stuart Russel & Peter Norvig, “Artificial Intelligence: A Modern Approach”, 4th edition, Pearson, 2020

# Learning Methods

## Unsupervised Learning

- Sebuah learning algorithm yang menerima sekumpulan data dan berusaha menemukan pola-pola di dalamnya.
- Contoh task: clustering

## Supervised Learning

- Sebuah learning algorithm yang menerima sekumpulan data dan dapat memetakan input ke output tertentu. Pada tahap training, learning algorithm menerima dataset pasangan input-output yang spesifik. Dataset ini digunakan untuk estimasi fungsi.
- Contoh task: classification, regression

# Learning Methods

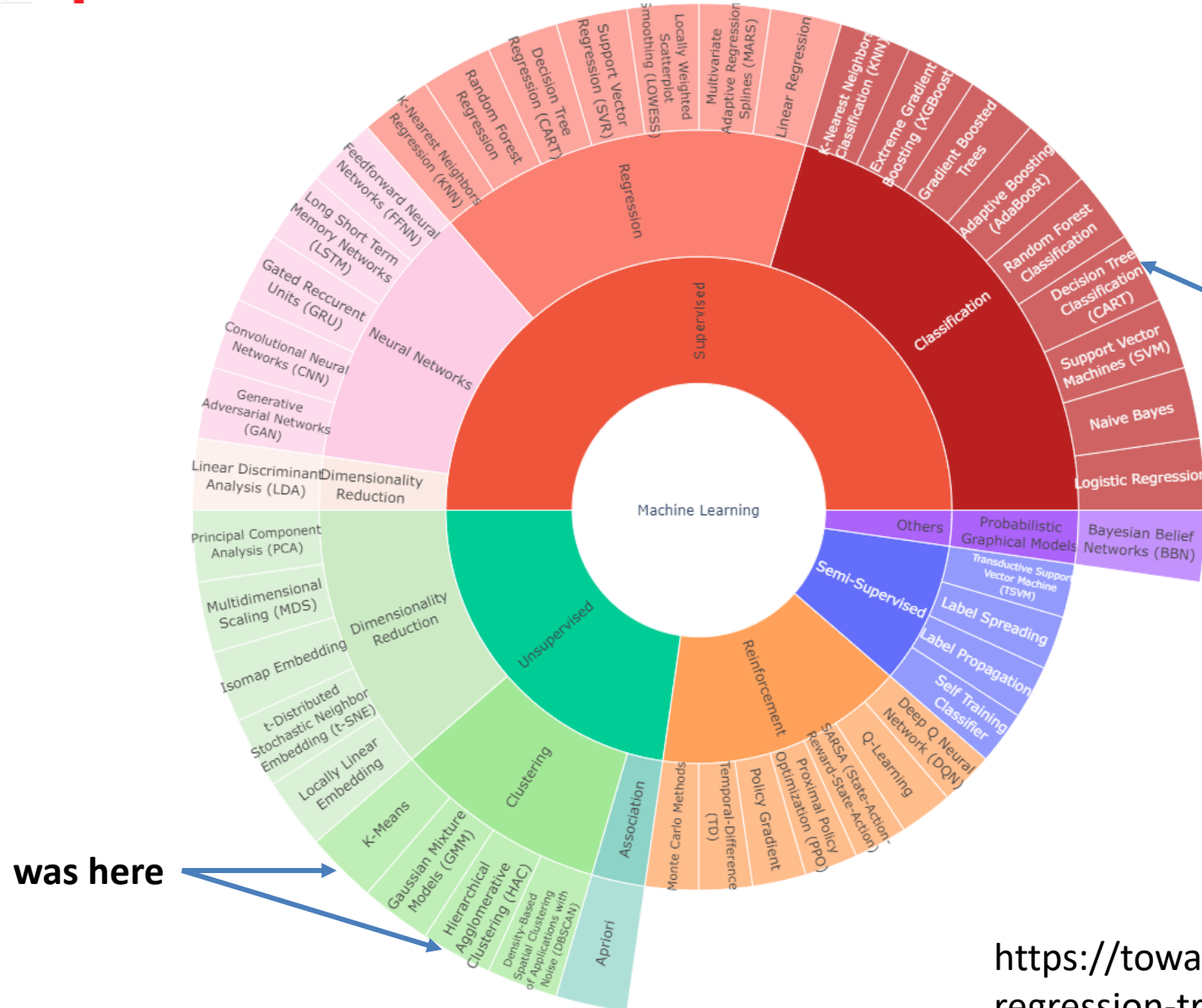
## Semi-supervised Learning

- Learning algorithm menerima beberapa (labeled) input yang memiliki pasangan output (baik akurat maupun noise). Kemudian, learning algorithm harus memberikan output untuk banyak input baru (unlabeled).
- Kesatuan dari supervised learning dan unsupervised learning

## Reinforcement Learning

- Learning algorithm menerima input data dan harus mengambil tindakan/memetakan output. Lalu, algoritma menerima reinforcement signal (berupa reward atau punishment mis. good, bad) sebagai akibat tindakan.
- Learning algorithm memodifikasi fungsi untuk memaksimalkan signal “good”.

# Learning Methods



<https://towardsdatascience.com/cart-classification-and-regression-trees-for-clean-but-powerful-models-cc89e60b7a85>



# Supervised Learning

- Prinsip dasar: Mempelajari fungsi atau aturan dari pasangan input-output atau “belajar dari pengalaman” (inductive learning).
- Diberikan sebuah training set (example)  $N$  berupa pasangan input-output:  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$ , di mana  $y_i$  dihasilkan dari suatu fungsi yang tidak diketahui  $y = f(x)$ . Supervised learning mencari fungsi hipotesis  $h$  yang mengaproksimasi  $f$ .
- Fungsi  $h$  yang bagus bisa memprediksi data yang belum dilihat pada saat belajar (training).
  - Jika domain output  $y$  diskrit: classification
  - Jika domain output  $y$  kontinu: regression

# Classification & Regression Trees

## Sumber:

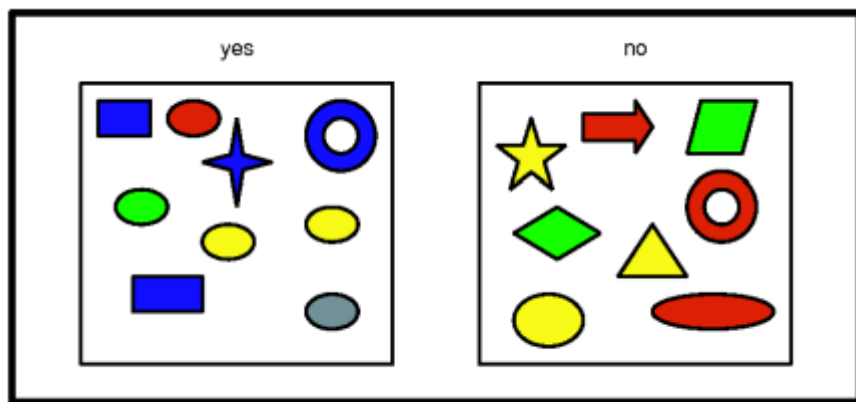
- Adila A. Krisnadhi, Slides Materi Pemelajaran Mesin, “CART and Random Forests”, Semester Genap 2020/2021
- Kevin P. Murphy, “Probabilistic Machine Learning: An Introduction”, MIT Press, 2021.
- Stuart Russel & Peter Norvig, “Artificial Intelligence: A Modern Approach”, 4th edition, Pearson, 2020Slide SC



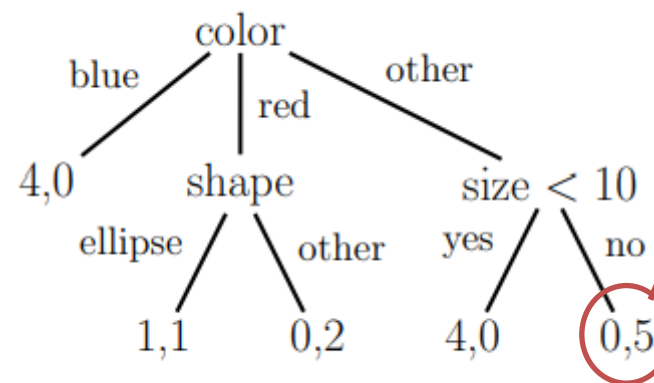
# CART

- Also known as decision trees
- Sebuah representasi dari kemungkinan fungsi hipotesis  $h$ , memprediksi nilai output variable ( $y_n$ ) berdasarkan input variable  $\mathbf{x}_n = (x_1, x_2, x_3, \dots, x_d)$
- Dapat dianggap sebagai sebuah **if...then** yang besar!
- Decision tree terdiri atas:
  - Internal node: representasi dari pengujian terhadap suatu nilai variable input
  - Himpunan edge dari suatu node: menyatakan kemungkinan nilai dari suatu variable input
  - Leaf node: memberikan nilai fungsi (output)

## Classification Tree

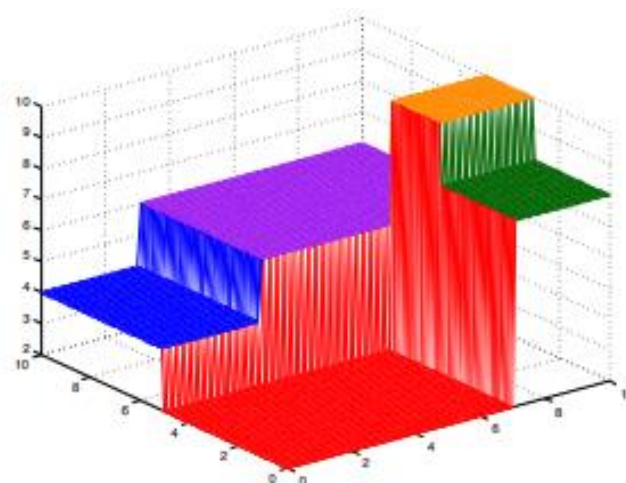
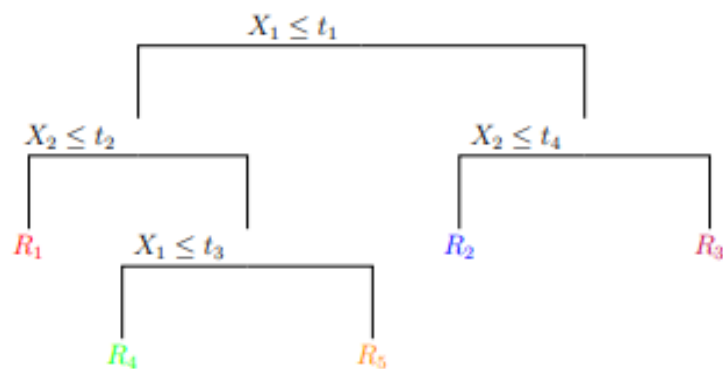


## CART



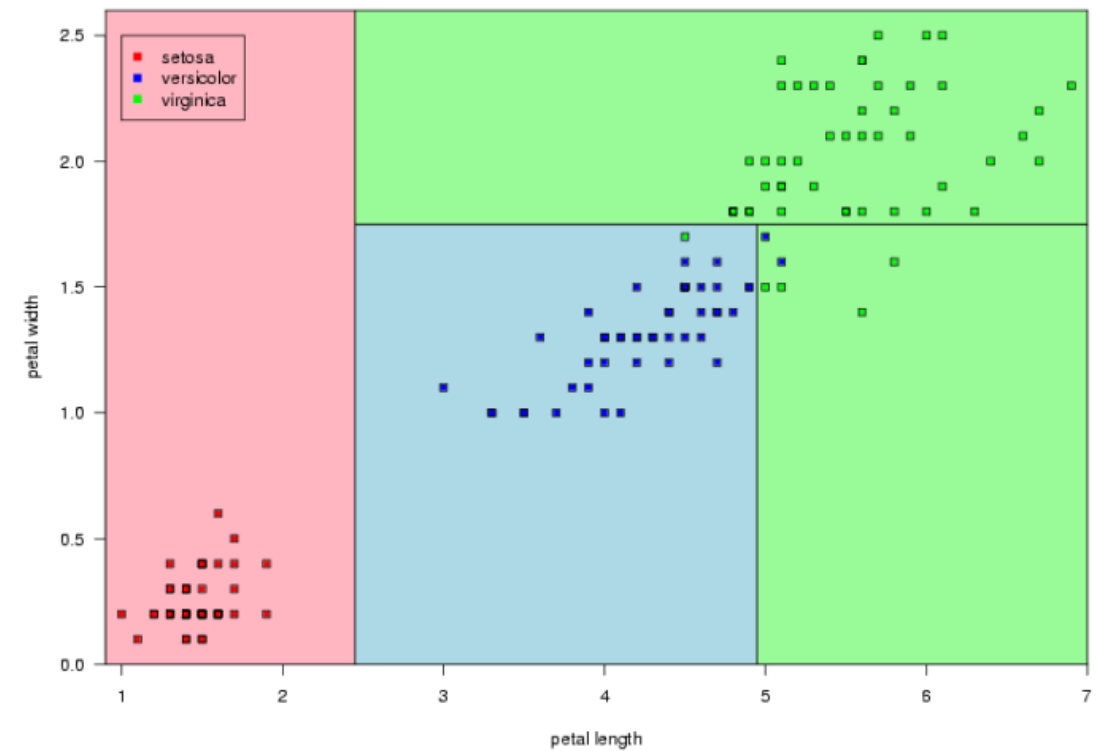
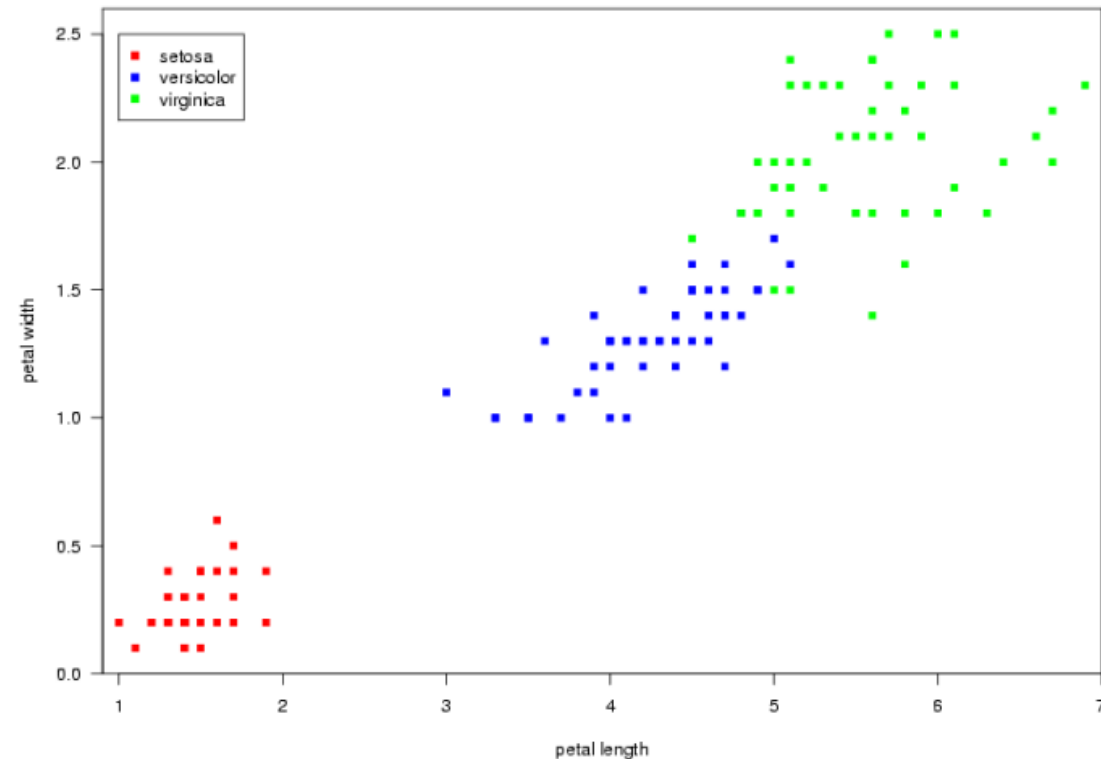
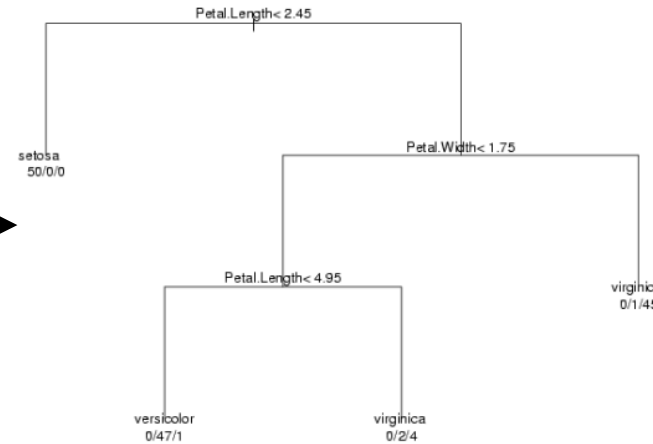
Menunjukkan jumlah objek yang masuk pada node tersebut, i.e. (jumlah objek kelas "Yes", jumlah objek kelas "No")

## Regression Tree



# CART: Iris Dataset

<https://rafalab.github.io/pages/649/section-11.pdf>



# Building a Decision Tree

Umumnya menggunakan pendekatan greedy

- Hunt's Algorithm (1966): *one of the earliest algorithms*, basis untuk algoritma lainnya
- **CART (1984)**
- ID3 (1986)
- C4.5/4.8/5.0 (1993)
- SLIQ, SPRINT (1996)

# Algoritma CART

$\mathbf{x}_n = (x_1, x_2, x_3, \dots, x_d)$  adalah input (example) dan  $j \in \{1, 2, 3, \dots, d\}$  adalah variable input (atribut).

Bila  $j$  variabel input kategorikal,  $\mathcal{T}_j$  adalah himpunan semua kemungkinan nilai diskret  $j$ .

Bila  $j$  variabel input numerik/kontinu,  $\mathcal{T}_j$  adalah himpunan semua kemungkinan nilai threshold  $j$ .

1. Pada node  $i$ ,  $\mathcal{D}_i = \{(\mathbf{x}_n, y_n) \in N_i\}$  adalah himpunan examples yang mencapai node  $i$
2. Untuk setiap variabel input  $j$  dan nilai  $t \in \mathcal{T}_j$ , lakukan binary split terhadap  $\mathcal{D}_i$  sehingga menghasilkan 2 subhimpunan  $\mathcal{D}_i^1(j, t)$  dan  $\mathcal{D}_i^2(j, t)$ , yaitu:
  - $\mathcal{D}_i^1(j, t) = \{(\mathbf{x}_n, y_n) \in N_i, x_j \leq t\}$  dan  $\mathcal{D}_i^2(j, t) = \{(\mathbf{x}_n, y_n) \in N_i, x_j > t\}$  jika  $j$  kontinu/numerik
  - $\mathcal{D}_i^1(j, t) = \{(\mathbf{x}_n, y_n) \in N_i, x_j = t\}$  dan  $\mathcal{D}_i^2(j, t) = \{(\mathbf{x}_n, y_n) \in N_i, x_j \neq t\}$  jika  $j$  kategorikal
3. Pilih binary split terhadap  $\mathcal{D}_i$  yang optimal, yaitu pasangan  $(j_i, t_i)$  yang memenuhi

$$(j_i, t_i) = \arg \min_{j \in \{1, 2, \dots, d\}} \min_{t \in \mathcal{T}_j} \left( \frac{|\mathcal{D}_i^1(j, t)|}{|\mathcal{D}_i|} \text{cost}(\mathcal{D}_i^1(j, t)) + \frac{|\mathcal{D}_i^2(j, t)|}{|\mathcal{D}_i|} \text{cost}(\mathcal{D}_i^2(j, t)) \right)$$

# Algoritma CART

4. Ketika sudah mendapat binary split yang optimal terhadap node  $i$  (i.e., menggunakan variabel input  $j_i$  dan nilai variabel input  $t_i$  sebagai threshold), buat 2 node anak  $i_1$  dan  $i_2$  sehingga  $\mathcal{D}_{i_1} = \mathcal{D}_i^1(j_i, t_i)$  dan  $\mathcal{D}_{i_2} = \mathcal{D}_i^2(j_i, t_i)$
5. Lakukan kembali langkah 1-4 sampai kondisi terminasi tercapai

## Pertanyaan:

- Apa definisi  $cost(\mathcal{D}_i)$ ?
- Bagaimana menentukan semua kemungkinan binary split pada sebuah node?
- Kapan algoritma CART berhenti?

# Fungsi Cost

- Regression Tree

$$cost(\mathcal{D}_i) = \frac{1}{|\mathcal{D}_i|} \sum_{n=1}^{|\mathcal{D}_i|} (y_n - \bar{y})^2$$

- dapat menggunakan mean-squared error (MSE) di mana  $\bar{y} = \frac{1}{|\mathcal{D}_i|} \sum_n y_n$  adalah mean dari nilai variabel output pada himpunan  $\mathcal{D}_i$

- Classification Tree

Bila  $*p_{i,c}$  adalah probabilitas node  $i$  mengandung examples berlabel kelas  $c$ , maka fungsi cost:

- Gini Index,

$$cost(\mathcal{D}_i) = \mathcal{G}_i = \sum_{c=1}^C p_{i,c}(1 - p_{i,c}) = \sum_{c=1}^C p_{i,c} - \sum_{c=1}^C p_{i,c}^2 = 1 - \sum_{c=1}^C p_{i,c}^2$$

- atau Entropy (deviance/impurity),

$$cost(\mathcal{D}_i) = H_i = \mathbb{H}(p_i) = - \sum_{c=1}^C p_{i,c} \log p_{i,c}$$

$$\begin{aligned} *p_{i,c} &= \frac{1}{|\mathcal{D}_i|} \sum_n \mathbb{I}(y_n = c) \\ \mathbb{I}(y_n = c) &= 1 \text{ jika } y_n = c, \\ \mathbb{I}(y_n = c) &= 0 \text{ otherwise} \end{aligned}$$

*Sebuah pure node adalah node yang mempunyai entropy 0 (hanya mengandung examples dengan nilai output sama)*



# How to Split

- Untuk variabel input  $j$  yang numerik/kontinu, buat list terurut (ascending) yang terdiri atas semua kemungkinan nilai  $j$  pada examples di  $\mathcal{D}_i$ , e.g.
  - 40, 48, 60, 72, 80, 90 adalah semua kemungkinan nilai  $j$  pada examples di  $\mathcal{D}_i$
  - Cari midpoint di antara nilai-nilai tersebut sebagai split point  $t$  sehingga  $\mathcal{T}_j = \{44, 54, 66, 76, 85\}$
  - Atau, gunakan nilai-nilai tersebut sebagai split point  $t$  sehingga  $\mathcal{T}_j = \{40, 48, 60, 72, 80, 90\}$
  - Atau, bagi interval  $[40, 90]$  menjadi  $k$  subinterval yang mempunyai panjang sama dan gunakan batas antar subinterval sebagai split point  $t$
- Untuk variabel input kategorikal,  $\mathcal{T}_j$  adalah himpunan semua kemungkinan nilai diskret  $j$  dari examples di  $\mathcal{D}_i$ 
  - Bila  $\mathcal{T}_j$  mengandung  $k$  nilai berbeda, maka akan ada  $k$  binary split yang berbeda
  - Atau, kita bisa melakukan multi-way split, i.e. membagi  $\mathcal{D}_i$  menjadi  $k$  subhimpunan yang berbeda

# In General...

Berdasarkan pendekatan greedy yang dijelaskan sebelumnya, maka proses membangun CART secara umum adalah:

- Mulai dari root node  $i$  sehingga  $\mathcal{D}_i$  diinisialisasi dengan semua examples pada training set
- Secara rekursif, membangun tree berdasarkan prosedur sebelumnya sampai membentuk leaf node
  - Leaf node adalah node yang mengandung examples dengan nilai variabel output yang sama

## Catatan:

Kita bisa memberlakukan pembatasan pada penggunaan variabel input untuk melakukan splitting pada node  $i$ , misalnya ketika atribut  $j$  sudah digunakan untuk split node  $i$ , maka  $j$  tidak digunakan lagi untuk splitting turunan dari node  $i$ . Jika hal ini terjadi, maka algoritma dapat berhenti ketika tidak ada variabel input yang bisa digunakan untuk splitting.

# CART Learning

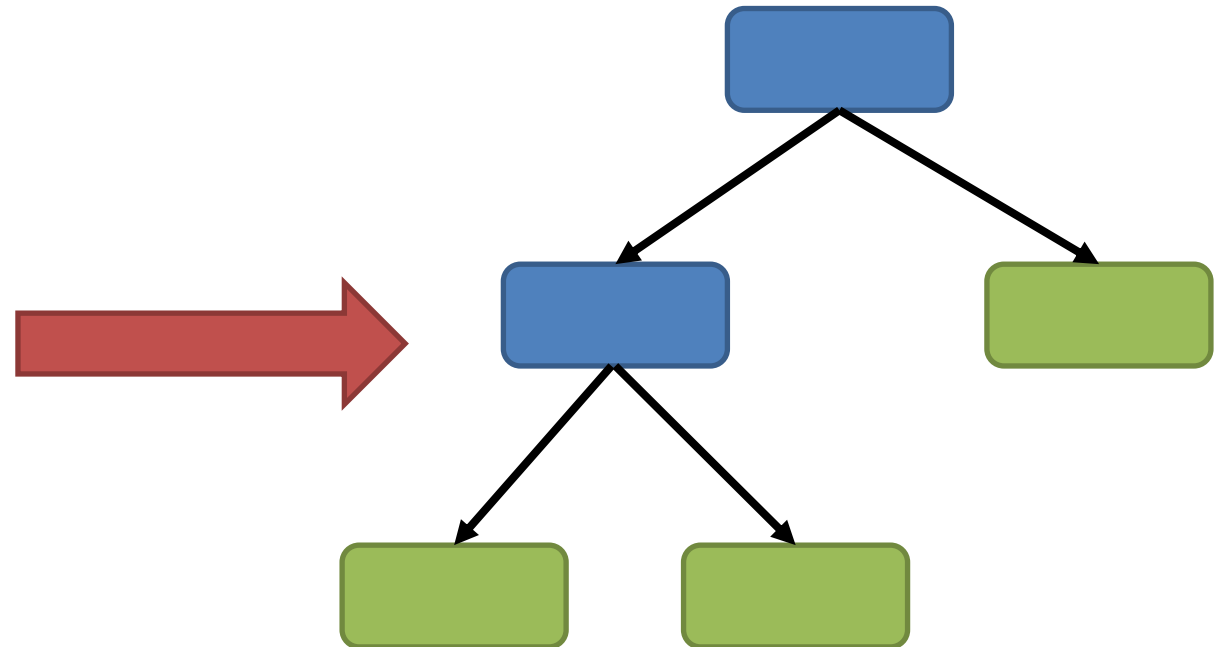
Sumber:

- Decision and Classification Trees, Clearly Explained!!! Video by StatQuest with Josh Starmer [StatQuest, 2021]

# Building a Classification Tree

- Berdasarkan examples/training set yang diberikan, bangunlah sebuah classification tree yang memprediksi apakah seseorang menyukai film “Ice Age”!

No.	Suka popcorn	Suka Minuman Soda	Umur	Suka “Ice Age”
1	Iya	Iya	7	Tidak
2	Iya	Tidak	12	Tidak
3	Tidak	Iya	18	Iya
4	Tidak	Iya	35	Iya
5	Iya	Iya	38	Iya
6	Iya	Tidak	50	Tidak
7	Tidak	Tidak	83	Tidak



# Building a Classification Tree

- Mulai dari root node yang mengandung semua examples, ada berapa kemungkinan binary split?

Suka Minuman Soda: 2 nilai  
1 binary split

No.	Suka popcorn	Suka Minuman Soda	Umur	Suka "Ice Age"
1	Iya	Iya	7	Tidak
2	Iya	Tidak	12	Tidak
3	Tidak	Iya	18	Iya
4	Tidak	Iya	35	Iya
5	Iya	Iya	38	Iya
6	Iya	Tidak	50	Tidak
7	Tidak	Tidak	83	Tidak

2 label kelas: "Iya" (Kelas 1),  
"Tidak" (Kelas 2)

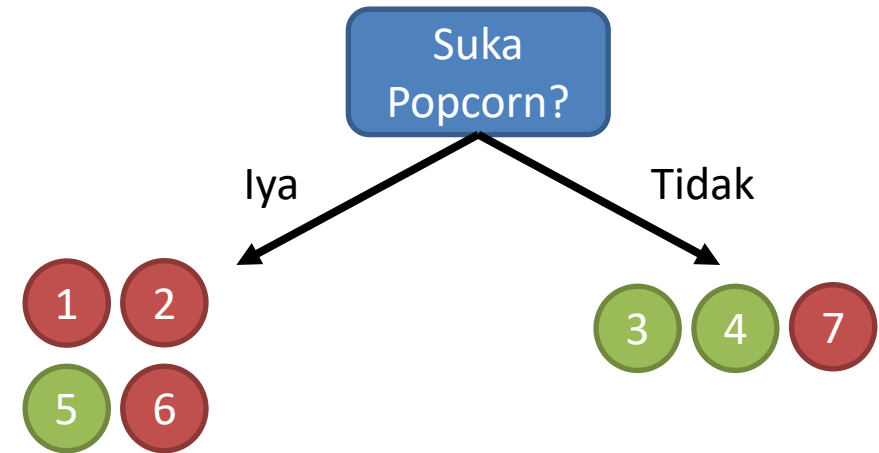
Suka popcorn: 2 nilai  
1 binary split

Umur: 7 nilai  
6 binary split

Ada 8 kemungkinan binary split yang dapat dilakukan pada root node.  
Hitung Cost tiap split!

# Split: Suka Popcorn

No.	Suka popcorn	...	Suka "Ice Age"
1	Iya	...	Tidak
2	Iya	...	Tidak
3	Tidak	...	Iya
4	Tidak	...	Iya
5	Iya	...	Iya
6	Iya	...	Tidak
7	Tidak	...	Tidak



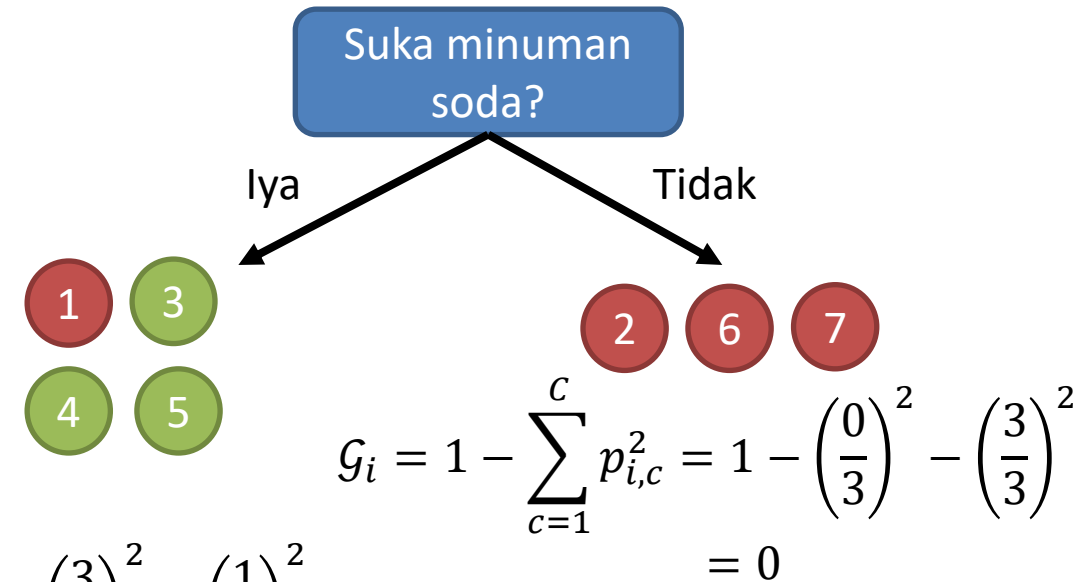
$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = 0.375$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = 0.444$$

$$\text{Maka } (j_i, t_i) = \frac{4}{7}(0.375) + \frac{3}{7}(0.444) = 0.405$$

# Split: Suka Minuman Soda

No.	Suka Minuman Soda	Suka "Ice Age"
1	Iya	Tidak
2	Tidak	Tidak
3	Iya	Iya
4	Iya	Iya
5	Iya	Iya
6	Tidak	Tidak
7	Tidak	Tidak



$$G_i = 1 - \sum_{c=1}^c p_{i,c}^2 = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = 0.375$$

$$\text{Maka } (j_i, t_i) = \frac{4}{7}(0.375) + \frac{3}{7}(0) = 0.214$$



# Split: Umur

No.	...	Umur	Suka "Ice Age"
1	...	7	Tidak
2	...	12	Tidak
3	...	18	Iya
4	...	35	Iya
5	...	38	Iya
6	...	50	Tidak
7	...	83	Tidak

Kebetulan data sudah terurut  
ascending berdasarkan umur, so...

Find split points  
{9.5, 15, 26.5, 36.5, 44, 66.5}

Hitung Gini index untuk setiap split point  $t$ , i.e. Ketika  $\leq t$  dan  $> t$

$$t = 9.5$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 = 0$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5$$

$$(j_i, t_i) = \frac{1}{7}(0) + \frac{6}{7}(0.5) = 0.429$$

$$t = 15$$

$$t = 26.5$$

$$t = 36.5$$

$$t = 44$$

$$t = 66.5$$

# Split: Umur

No.	...	Umur	Suka "Ice Age"
1	...	7	Tidak
2	...	12	Tidak
3	...	18	Iya
4	...	35	Iya
5	...	38	Iya
6	...	50	Tidak
7	...	83	Tidak

Kebetulan data sudah terurut  
ascending berdasarkan umur, so...

Find split points  
{9.5, 15, 26.5, 36.5, 44, 66.5}

Hitung Gini index untuk setiap split point  $t$ , i.e. Ketika  $\leq t$  dan  $> t$

$$t = 9.5 \quad (j_i, t_i) = 0.429$$

$$t = 15$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.48$$

$$(j_i, t_i) = \frac{2}{7}(0) + \frac{5}{7}(0.48) = 0.343$$

$$t = 26.5$$

$$t = 36.5$$

$$t = 44$$

$$t = 66.5$$

# Split: Umur

No.	...	Umur	Suka "Ice Age"
1	...	7	Tidak
2	...	12	Tidak
3	...	18	Iya
4	...	35	Iya
5	...	38	Iya
6	...	50	Tidak
7	...	83	Tidak

Kebetulan data sudah terurut  
ascending berdasarkan umur, so...

Find split points  
{9.5, 15, 26.5, 36.5, 44, 66.5}

Hitung Gini index untuk setiap split point  $t$ , i.e. Ketika  $\leq t$  dan  $> t$

$$t = 9.5 \quad (j_i, t_i) = 0.429$$

$$t = 15 \quad (j_i, t_i) = 0.343$$

$$t = 26.5$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

$$(j_i, t_i) = \frac{3}{7}(0.444) + \frac{4}{7}(0.5) = 0.476$$

$$t = 36.5$$

$$t = 44$$

$$t = 66.5$$

# Split: Umur

No.	...	Umur	Suka "Ice Age"
1	...	7	Tidak
2	...	12	Tidak
3	...	18	Iya
4	...	35	Iya
5	...	38	Iya
6	...	50	Tidak
7	...	83	Tidak

Kebetulan data sudah terurut  
ascending berdasarkan umur, so...

Find split points  
{9.5, 15, 26.5, 36.5, 44, 66.5}

Hitung Gini index untuk setiap split point  $t$ , i.e. Ketika  $\leq t$  dan  $> t$

$$t = 9.5 \quad (j_i, t_i) = 0.429$$

$$t = 15 \quad (j_i, t_i) = 0.343$$

$$t = 26.5 \quad (j_i, t_i) = 0.476$$

$$t = 36.5$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

$$(j_i, t_i) = \frac{4}{7}(0.5) + \frac{3}{7}(0.444) = 0.476$$

$$t = 44$$

$$t = 66.5$$

# Split: Umur

No.	...	Umur	Suka "Ice Age"
1	...	7	Tidak
2	...	12	Tidak
3	...	18	Iya
4	...	35	Iya
5	...	38	Iya
6	...	50	Tidak
7	...	83	Tidak

Kebetulan data sudah terurut  
ascending berdasarkan umur, so...

Find split points

{9.5, 15, 26.5, 36.5, 44, 66.5}

Hitung Gini index untuk setiap split point  $t$ , i.e. Ketika  $\leq t$  dan  $> t$

$$t = 9.5 \quad (j_i, t_i) = 0.429$$

$$t = 15 \quad (j_i, t_i) = 0.343$$

$$t = 26.5 \quad (j_i, t_i) = 0.476$$

$$t = 36.5 \quad (j_i, t_i) = 0.476$$

$$t = 44$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.48$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$(j_i, t_i) = \frac{5}{7}(0.48) + \frac{2}{7}(0) = 0.343$$

$$t = 66.5$$

# Split: Umur

No.	...	Umur	Suka "Ice Age"
1	...	7	Tidak
2	...	12	Tidak
3	...	18	Iya
4	...	35	Iya
5	...	38	Iya
6	...	50	Tidak
7	...	83	Tidak

Kebetulan data sudah terurut  
ascending berdasarkan umur, so...

Find split points  
{9.5, 15, 26.5, 36.5, 44, 66.5}

Hitung Gini index untuk setiap split point  $t$ , i.e. Ketika  $\leq t$  dan  $> t$

$$t = 9.5 \quad (j_i, t_i) = 0.429$$

$$t = 15 \quad (j_i, t_i) = 0.343$$

$$t = 26.5 \quad (j_i, t_i) = 0.476$$

$$t = 36.5 \quad (j_i, t_i) = 0.476$$

$$t = 44 \quad (j_i, t_i) = 0.343$$

$$t = 66.5$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5$$


$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 = 0$$


$$(j_i, t_i) = \frac{6}{7} (0.5) + \frac{1}{7} (0) = 0.429$$


Best split untuk umur adalah ketika  $t = 15$  atau  $44$  dengan  $0.343$

# Best Split at Root?

No.	Suka popcorn	Suka Minuman Soda	Umur	Suka "Ice Age"
1	Iya	Iya	7	Tidak
2	Iya	Tidak	12	Tidak
3	Tidak	Iya	18	Iya
4	Tidak	Iya	35	Iya
5	Iya	Iya	38	Iya
6	Iya	Tidak	50	Tidak
7	Tidak	Tidak	83	Tidak

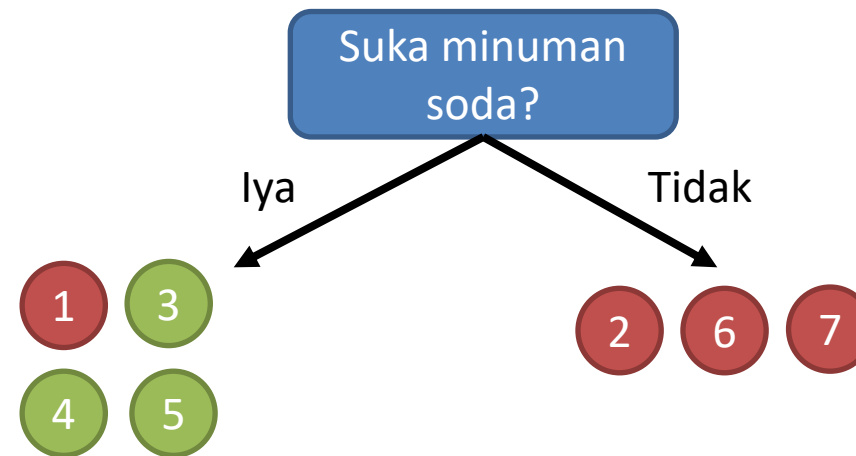
  
0.405

  
0.214

  
0.343

Pilih nilai minimum, yaitu 0.214.

Split point di root menggunakan "Suka Minuman Soda"



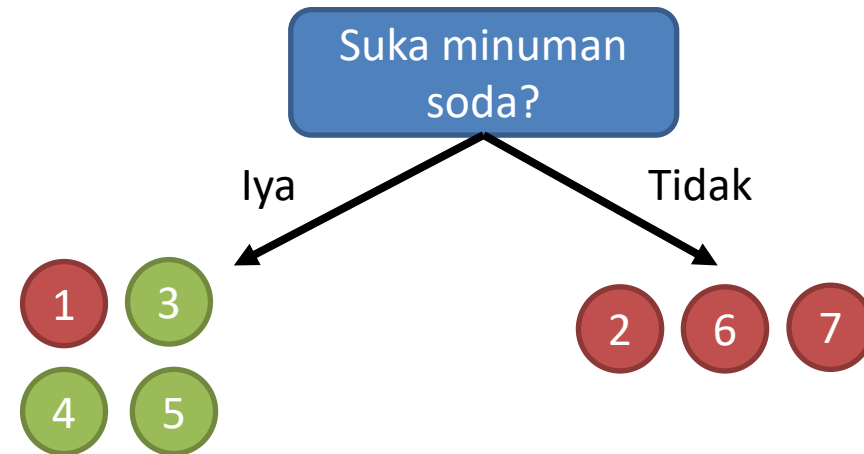
.... But wait, there's more!



# Next Splits

No.	Suka popcorn	Suka Minuman Soda	Umur	Suka "Ice Age"
1	Iya	Iya	7	Tidak
2	Iya	Tidak	12	Tidak
3	Tidak	Iya	18	Iya
4	Tidak	Iya	35	Iya
5	Iya	Iya	38	Iya
6	Iya	Tidak	50	Tidak
7	Tidak	Tidak	83	Tidak

Lakukan langkah 1-4 pada masing-masing node anak



Tapi...

Node pada cabang “Tidak” mengandung examples berlabel “Tidak Suka Ice Age” saja. Oleh karena itu, jadikan node anak ini sebagai leaf node dengan label “Tidak Suka Ice Age”.

Node cabang “Iya” masih mengandung examples dengan kelas berbeda. Lakukan split pada node tersebut!

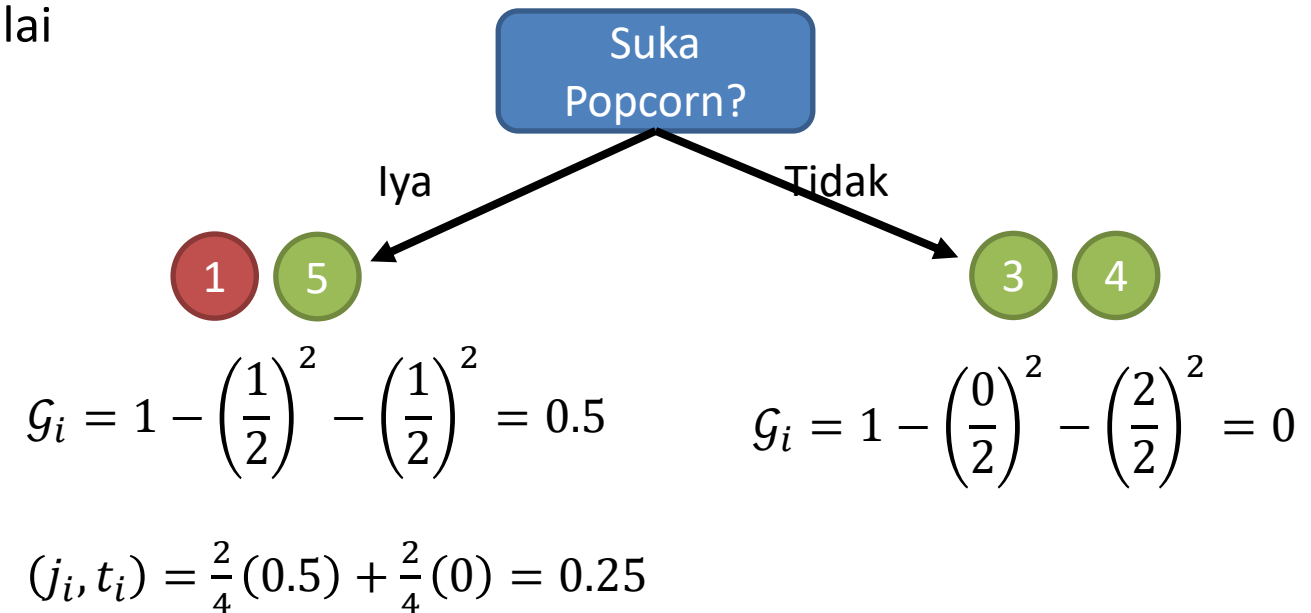
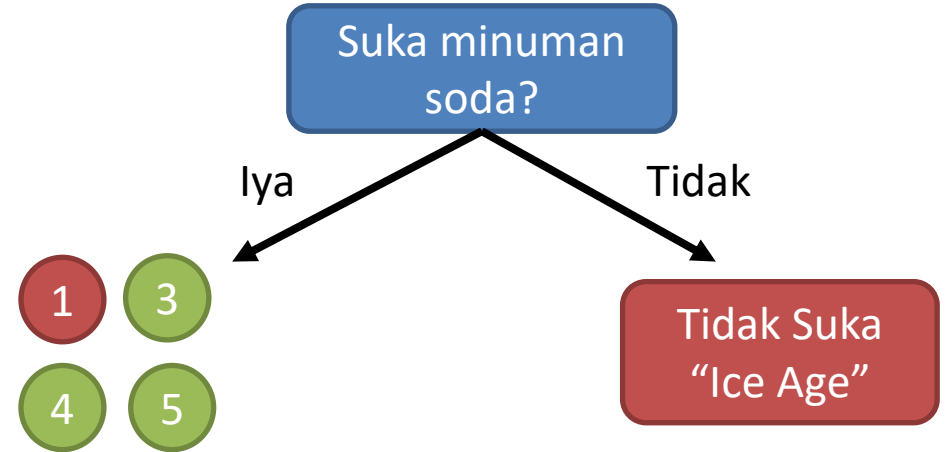
# Next Splits

No.	Suka popcorn	Suka Minuman Soda	Umur	Suka "Ice Age"
1	Iya	Iya	7	Tidak
3	Tidak	Iya	18	Iya
4	Tidak	Iya	35	Iya
5	Iya	Iya	38	Iya

Hitung split cost untuk setiap variabel input dengan nilai thresholdnya!

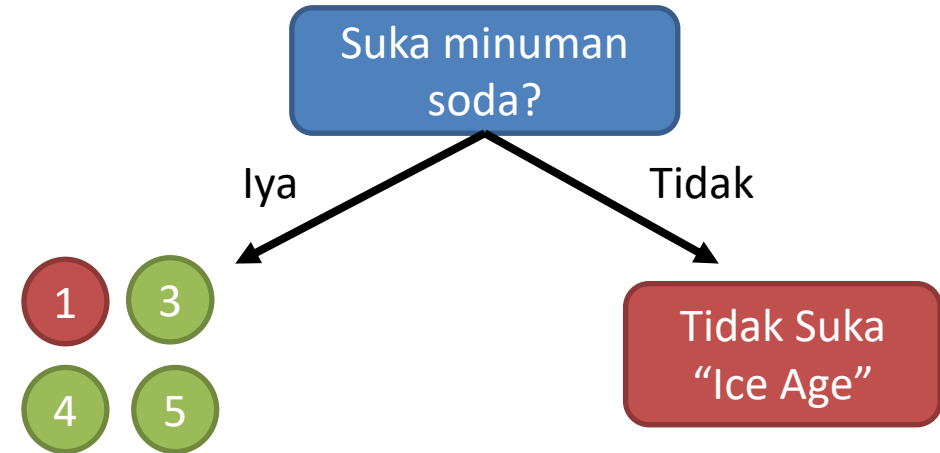
1) Suka popcorn

No.	Suka popcorn	...	Suka "Ice Age"
1	Iya	...	Tidak
3	Tidak	...	Iya
4	Tidak	...	Iya
5	Iya	...	Iya



# Next Splits

No.	Suka popcorn	Suka Minuman Soda	Umur	Suka "Ice Age"
1	Iya	Iya	7	Tidak
3	Tidak	Iya	18	Iya
4	Tidak	Iya	35	Iya
5	Iya	Iya	38	Iya



Hitung split cost untuk setiap variabel input dengan nilai thresholdnya!

2) Suka Minuman Soda: Tidak perlu, karena variabel kategorikal **yang hanya mempunyai 2 kemungkinan nilai** dan sudah ditanyakan nilainya, bila ditanya lagi pada cabang ini, data tetap tidak akan terpisah berdasarkan label kelas. So, ...langsung hitung untuk variabel Umur!

\*walau IRL bisa saja algoritma tetap menghitung Gini Index-nya

# Next Splits

Hitung Gini index untuk setiap split point  $t$ , i.e. Ketika  $\leq t$  dan  $> t$

$t = 12.5$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 = 0$$

$$G_i = 1 - \sum_{c=1}^C p_{i,c}^2 = 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

$$(j_i, t_i) = 0$$

$t = 26.5$

$t = 36.5$

Perhatikan untuk  $t$  lainnya. Bila split dilakukan pada nilai-nilai tersebut, minimal ada 1 node anak yang bercampur (impure node). Oleh karena itu, yang pasti terpilih adalah split dengan  $t = 12.5$

No.	Suka popcorn	Suka Minuman Soda	Umur	Suka "Ice Age"
1	Iya	Iya	7	Tidak
3	Tidak	Iya	18	Iya
4	Tidak	Iya	35	Iya
5	Iya	Iya	38	Iya

3) Umur

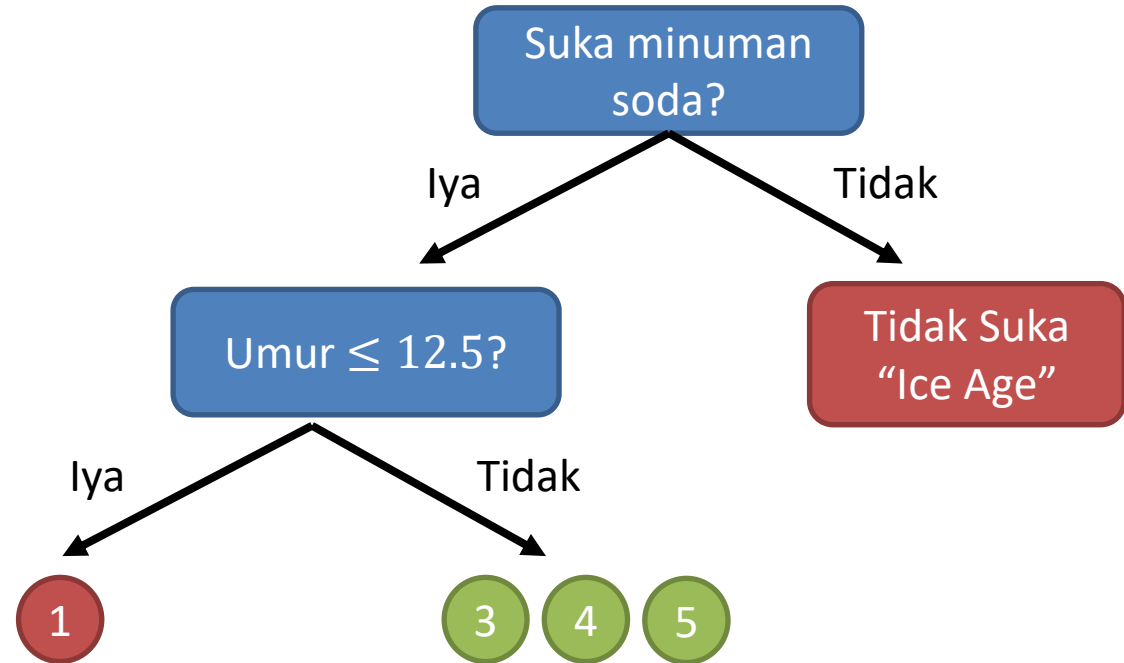
Find split points = {12.5, 26.5, 36.5}

Catatan: IRL, algoritma bisa saja tetap menghitung nilai untuk setiap split point  $t$

# Next Splits

No.	Suka popcorn	Suka Minuman Soda	Umur	Suka "Ice Age"
1	Iya	Iya	7	Tidak
3	Tidak	Iya	18	Iya
4	Tidak	Iya	35	Iya
5	Iya	Iya	38	Iya

Arrows from the 'Suka popcorn' and 'Umur' columns point to the values 0.25 and 0 respectively.



Pilih Umur dengan  $t = 12.5$  sebagai split point!

Perhatikan node anak, karena kedua-duanya pure node, jadikan sebagai leaf node!

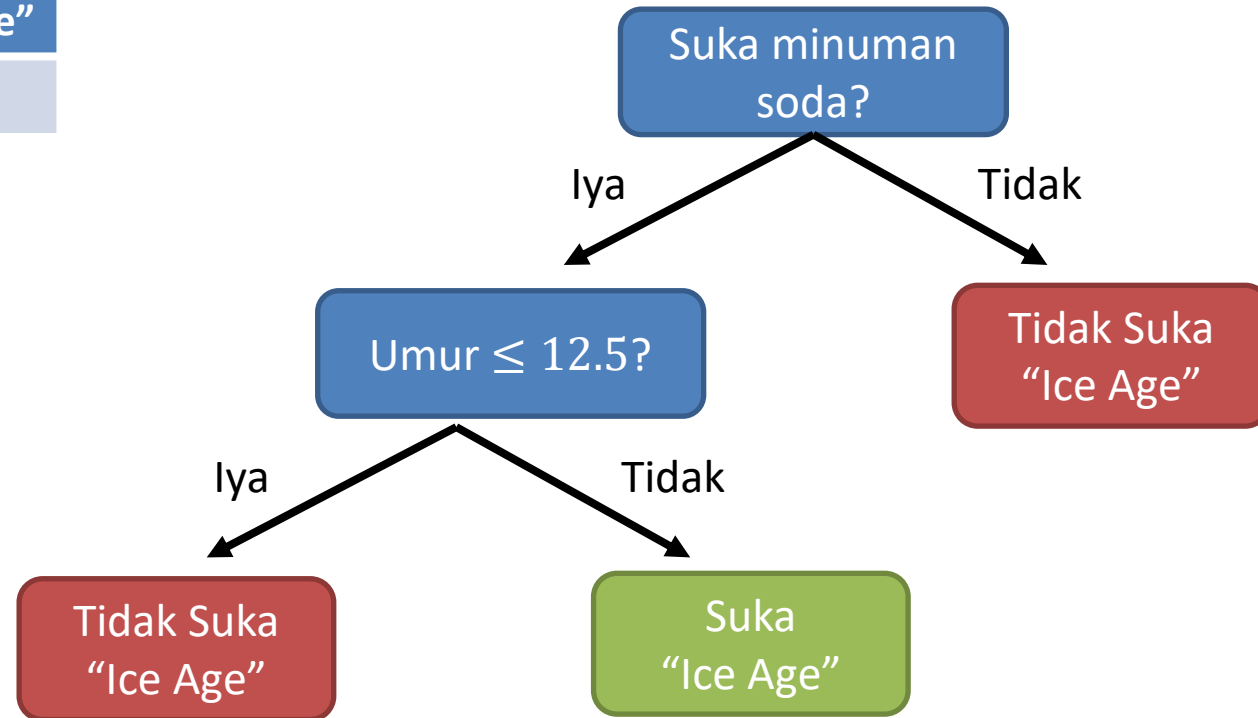
Node anak dari cabang "Iya" menjadi leaf node "Tidak Suka Ice Age"

Node anak dari cabang "Tidak" menjadi leaf node "Suka Ice Age"

# Hasil Akhir

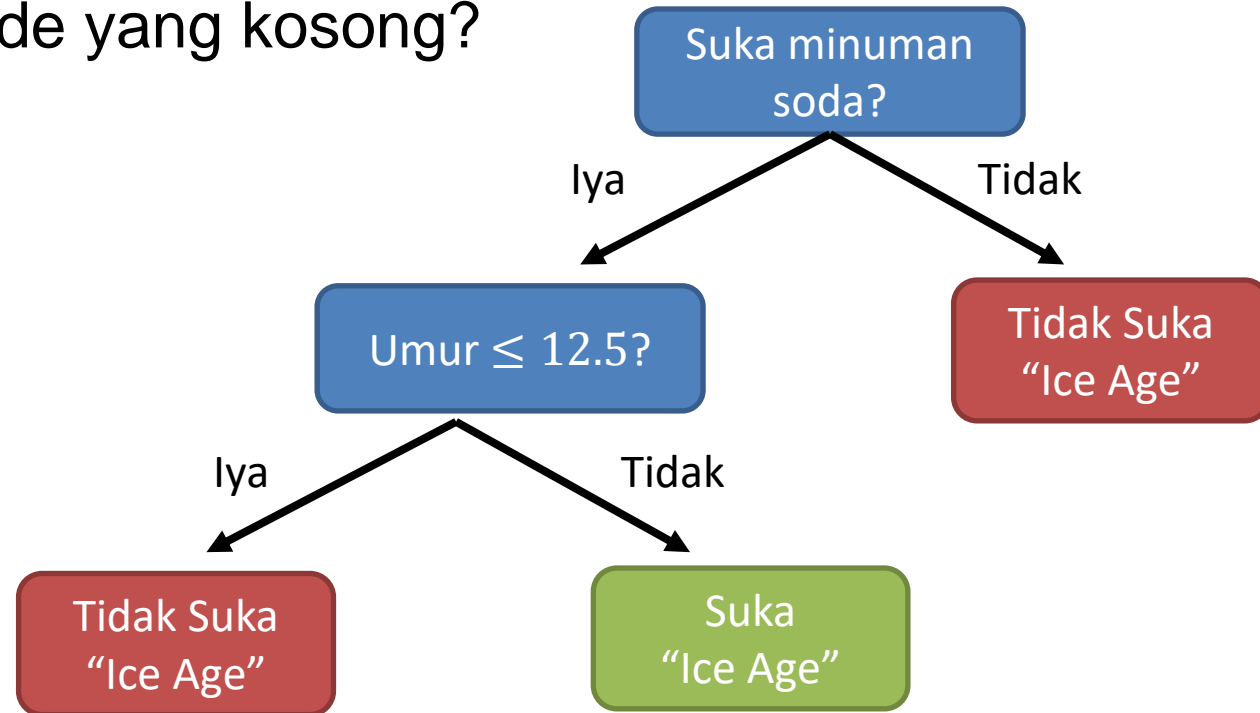
Mari kita prediksi data berikut, apakah nilai outputnya?

Suka popcorn	Suka Minuman Soda	Umur	Suka "Ice Age"
Iya	Iya	31	?



# Question

- Untuk contoh ini, seandainya node-node leaf di bawah masih impure, apakah masih ada variabel input yang bisa dijadikan sebagai splitting point? **Bila ternyata tidak ada variabel yang bisa dijadikan splitting point lagi, bagaimana caranya CART memberi label?**
- Bagaimana bila ada leaf node yang kosong?





# Building a Regression Tree

Pada dasarnya proses membangun regression tree mirip dengan membangun classification tree, tetapi ada perbedaan:

1. Nilai output adalah sebuah bilangan (variabel kontinu)
2. Fungsi cost yang digunakan:

$$cost(\mathcal{D}_i) = \frac{1}{|\mathcal{D}_i|} \sum_{n=1}^{|\mathcal{D}_i|} (y_n - \bar{y})^2$$

3. Bila tidak ada variabel input lain yang bisa digunakan untuk splitting sedangkan node masih mengandung **examples dengan nilai output berbeda (cost > 0)**, jadikan node sebagai leaf yang mengembalikan rata-rata nilai variabel output examples pada node tersebut.

Selain itu... sama!

Mari melihat contoh berikut...

# Building a Regression Tree

#	Nonton Video Tutorial?	Lab Lengkap?	Ujian
1	Semua	Iya	74
2	Sebagian	Tidak	23
3	Semua	Iya	61
4	Semua	Iya	74
5	Sebagian	Tidak	25
6	Semua	Iya	61
7	Sebagian	Iya	54
8	Sebagian	Tidak	42
9	Sebagian	Iya	55
10	Semua	Iya	75
11	Sebagian	Tidak	13
12	Semua	Iya	73
13	Sebagian	Tidak	31
14	Sebagian	Tidak	12
15	Sebagian	Tidak	11

Kita mulai dengan contoh yang lebih sederhana, menggunakan variabel input kategorikal.

Buatlah regression tree yang dapat memprediksi nilai ujian seorang mahasiswa bila dilihat dari frekuensi menonton video tutorial dan pengumpulan tugas lab!

# Building a Regression Tree

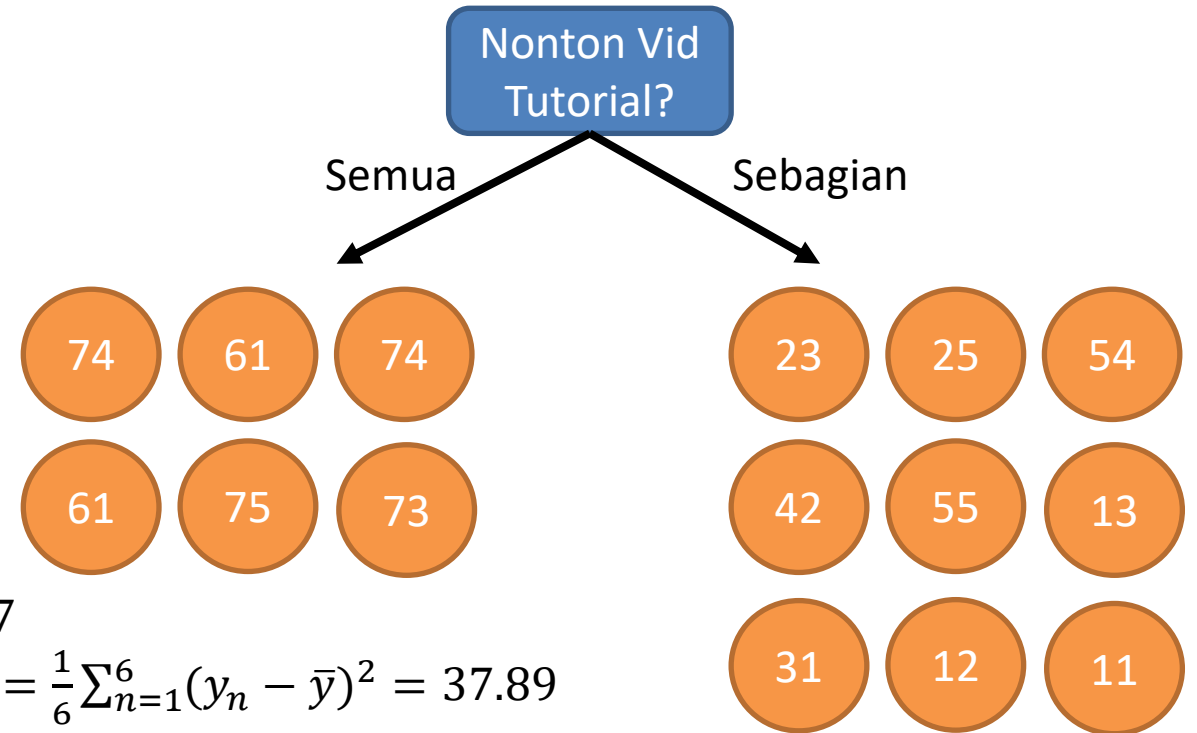
#	Nonton Video Tutorial?	Lab Lengkap?	Ujian
1	Semua	Iya	74
2	Sebagian	Tidak	23
3	Semua	Iya	61
4	Semua	Iya	74
5	Sebagian	Tidak	25
6	Semua	Iya	61
7	Sebagian	Iya	54
8	Sebagian	Tidak	42
9	Sebagian	Iya	55
10	Semua	Iya	75
11	Sebagian	Tidak	13
12	Semua	Iya	73
13	Sebagian	Tidak	31
14	Sebagian	Tidak	12
15	Sebagian	Tidak	11

Mulai dari root node yang mengandung semua examples, ada berapa kemungkinan binary split?

2 binary split!

# Split: Nonton Video Tutorial?

#	Nonton Video Tutorial?	Ujian
1	Semua	74
2	Sebagian	23
3	Semua	61
4	Semua	74
5	Sebagian	25
6	Semua	61
7	Sebagian	54
8	Sebagian	42
9	Sebagian	55
10	Semua	75
11	Sebagian	13
12	Semua	73
13	Sebagian	31
14	Sebagian	12
15	Sebagian	11



$$\bar{y} = 69.67$$

$$cost(\mathcal{D}_i) = \frac{1}{6} \sum_{n=1}^6 (y_n - \bar{y})^2 = 37.89$$

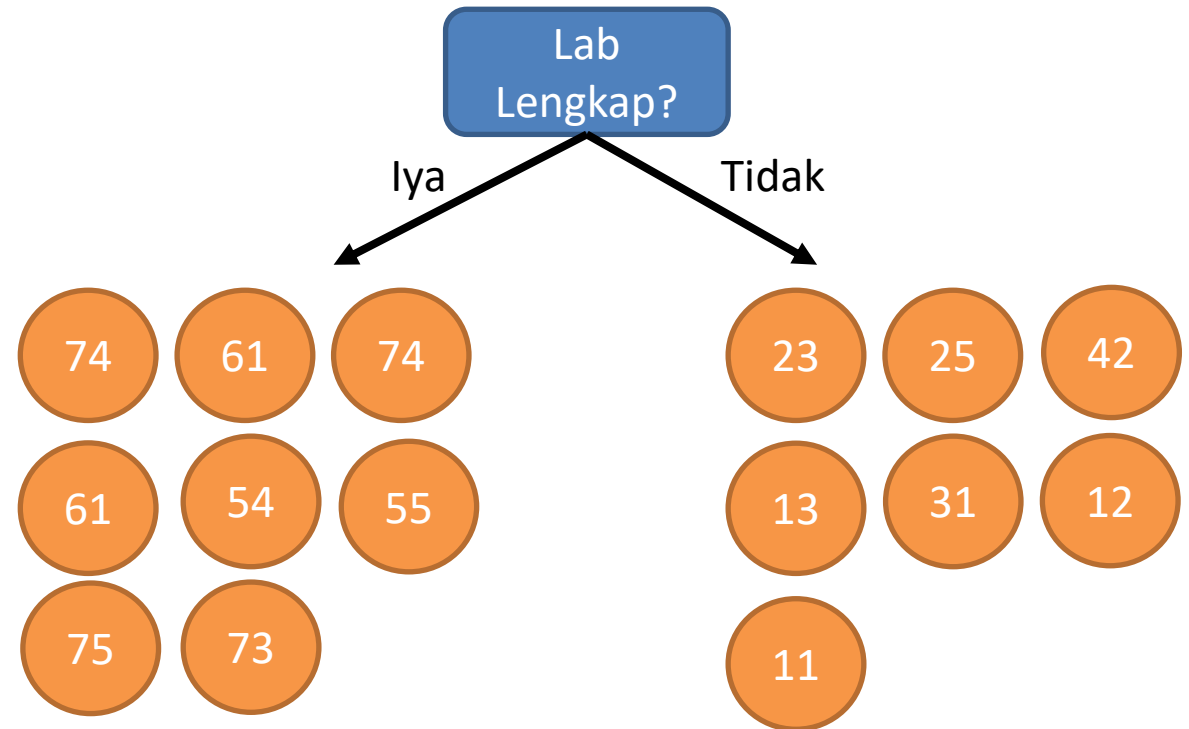
$$\bar{y} = 29.56$$

$$cost(\mathcal{D}_i) = \frac{1}{9} \sum_{n=1}^9 (y_n - \bar{y})^2 = 265.8$$

$$(j_i, t_i) = \frac{6}{15} (37.89) + \frac{9}{15} (265.8) = 174.636$$

# Split: Lab Lengkap?

#	Lab Lengkap?	Ujian
1	Iya	74
2	Tidak	23
3	Iya	61
4	Iya	74
5	Tidak	25
6	Iya	61
7	Iya	54
8	Tidak	42
9	Iya	55
10	Iya	75
11	Tidak	13
12	Iya	73
13	Tidak	31
14	Tidak	12
15	Tidak	11



$$\bar{y} = 65.88$$

$$\text{cost}(\mathcal{D}_i) = \frac{1}{8} \sum_{n=1}^8 (y_n - \bar{y})^2 = 71.61$$

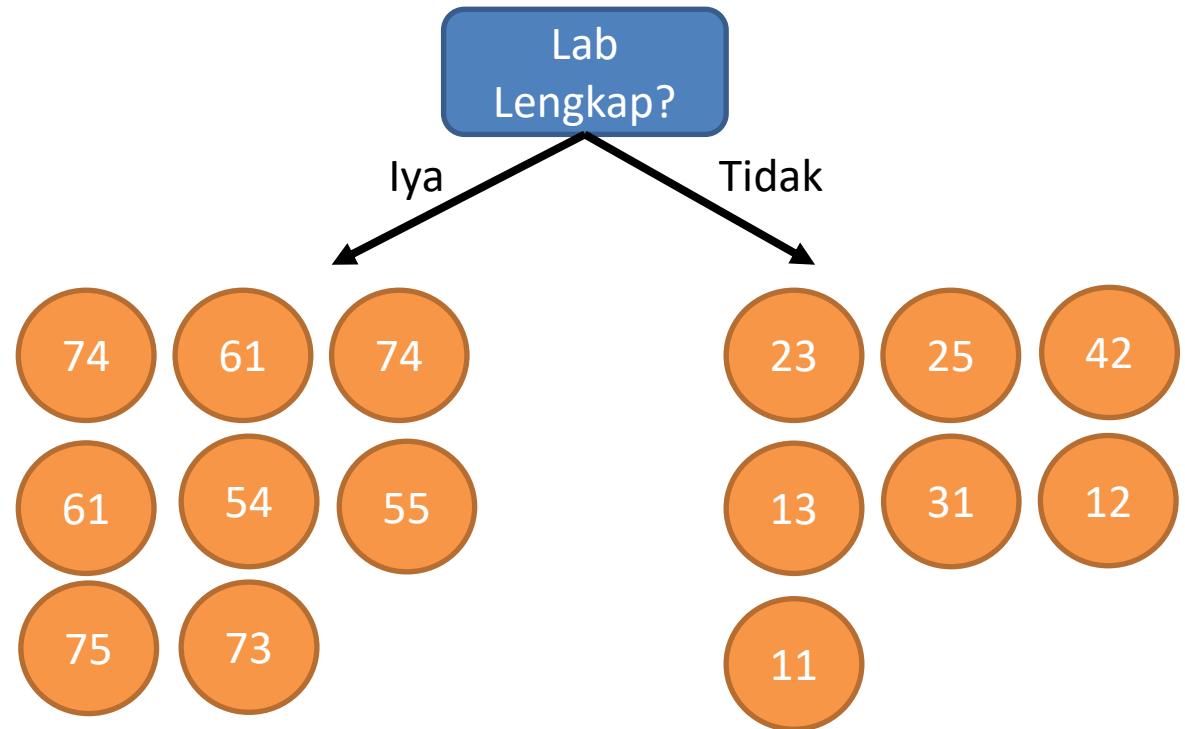
$$\bar{y} = 22.43$$

$$\text{cost}(\mathcal{D}_i) = \frac{1}{7} \sum_{n=1}^7 (y_n - \bar{y})^2 = 113.1$$

$$(j_i, t_i) = \frac{8}{15} (71.61) + \frac{7}{15} (113.1) = 90.97$$

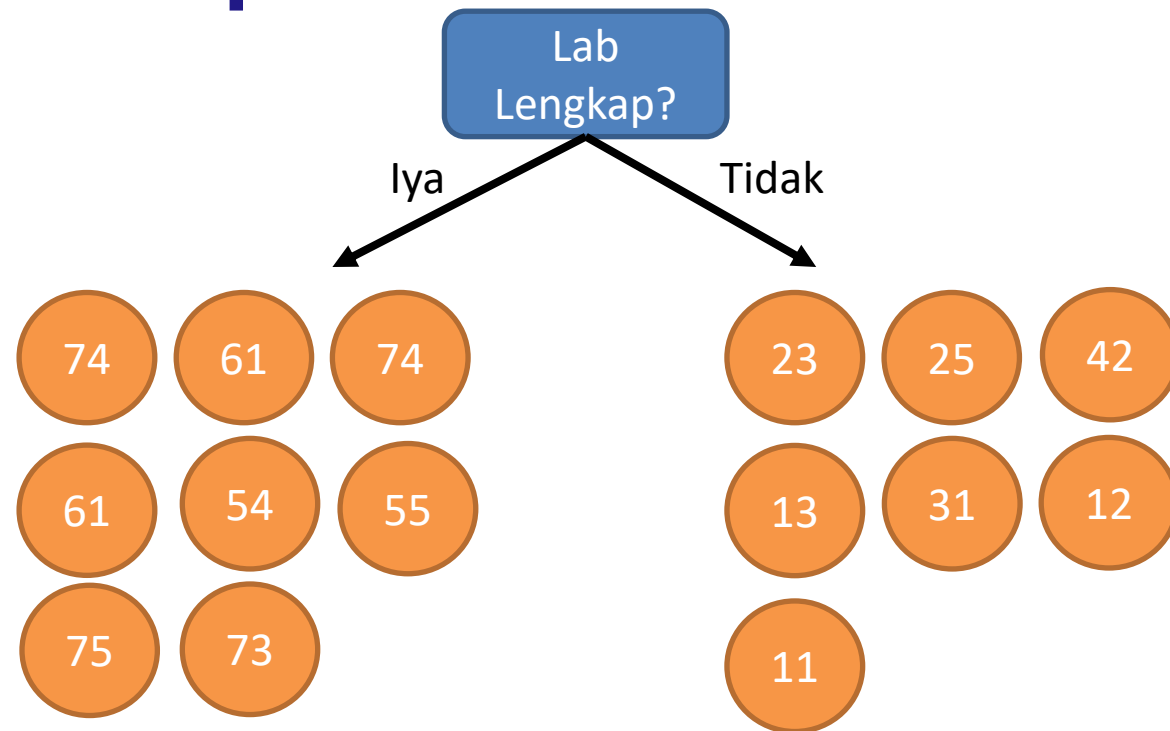
# Next Split

- Karena nilai Lab Lengkap lebih kecil, gunakan variabel ini sebagai splitting point di root.
- Tentukan splitting point pada node-node anak!



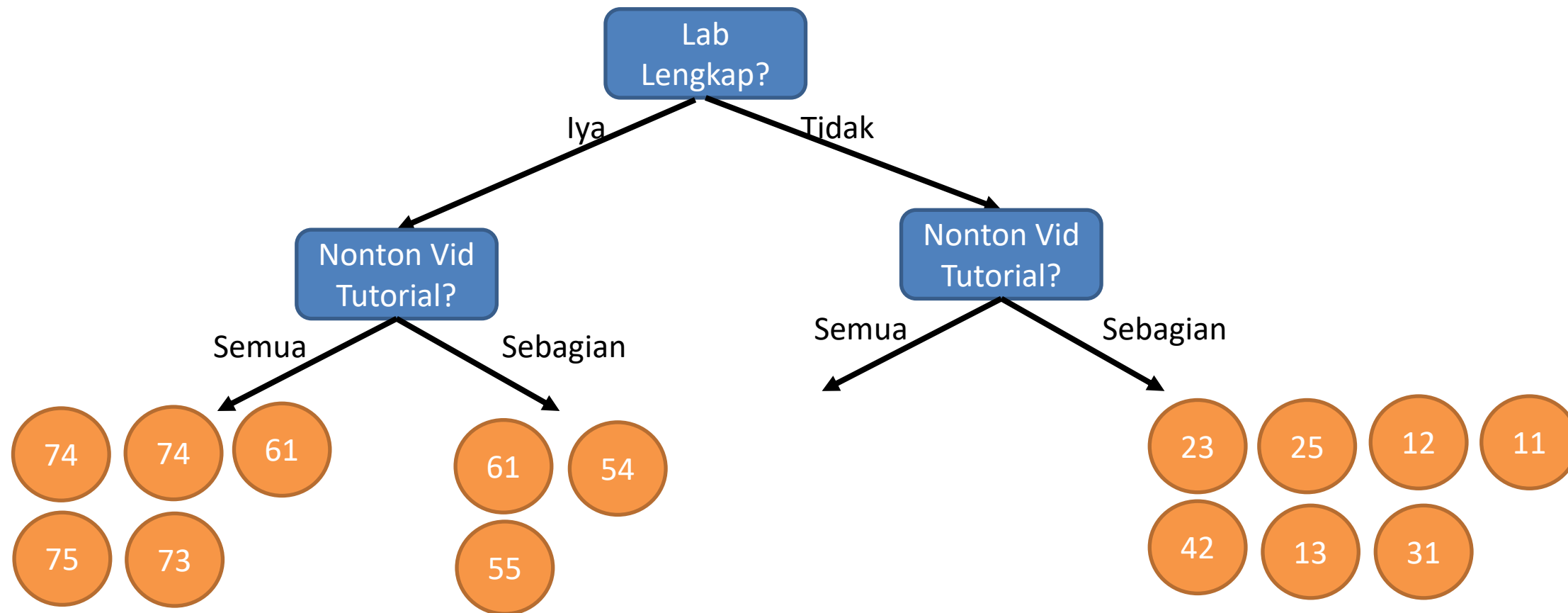
# Next Split

#	Nonton Video Tutorial?	Lab Lengkap?	Ujian
1	Semua	Iya	74
2	Sebagian	Tidak	23
3	Semua	Iya	61
4	Semua	Iya	74
5	Sebagian	Tidak	25
6	Semua	Iya	61
7	Sebagian	Iya	54
8	Sebagian	Tidak	42
9	Sebagian	Iya	55
10	Semua	Iya	75
11	Sebagian	Tidak	13
12	Semua	Iya	73
13	Sebagian	Tidak	31
14	Sebagian	Tidak	12
15	Sebagian	Tidak	11



Perhatikan training set, semua variabel input kategorikal dan **hanya mempunyai 2 kemungkinan nilai**, jadi hanya bisa digunakan sekali pada satu path. Satu-satunya splitting yang terjadi adalah dengan menggunakan variabel “Nonton Video Tutorial” pada kedua node anak.

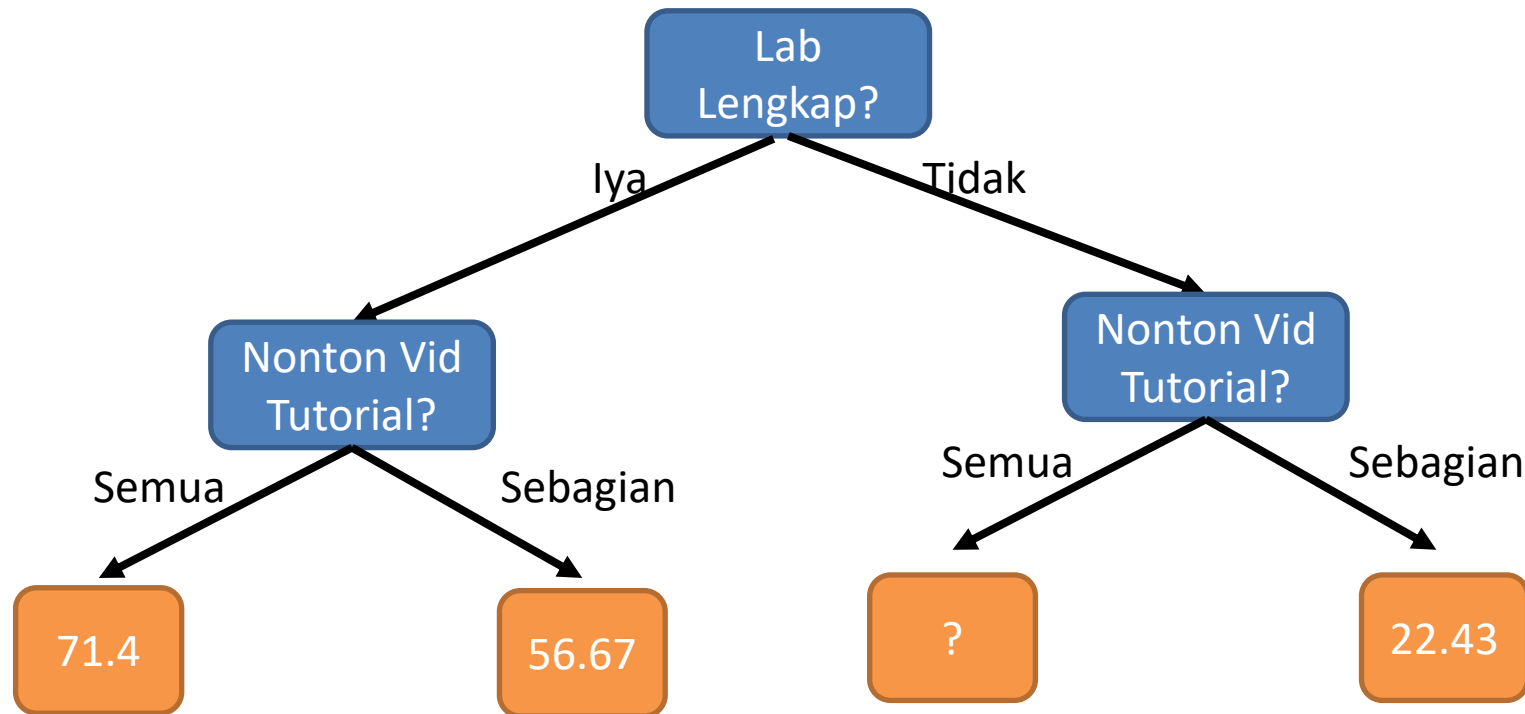
# Next Split



- Tidak ada variabel input lain yang bisa dijadikan split point.
- Semua node anak dijadikan leaf node dengan label  $\bar{y}$



# Question



- Bagaimana proses splitting terjadi bila ada tambahan variabel input numerik “Kuis”?
- Bagaimana bila ada leaf node kosong pada regression tree?

# Parameters & Hyperparameters

- Parameters: bagian dari model yang di-*learn* atau diestimasi selama proses training
  - Contoh: centroid tiap cluster dalam k-means clustering
- Hyperparameters: bagian dari model yang mengontrol proses learning/training, akibatnya akan mempunyai efek pada parameter hasil training sebuah model
  - Contoh: nilai k (banyaknya cluster) dalam k-means clustering
- Hyperparameters dapat kita tentukan sendiri atau berdasarkan hasil testing (*ideally cross-validation*)

# CART: Hyperparameters

- Hyperparameter decision tree yang dapat di-tuning:
  - max\_depth: kedalaman maksimum tree
  - min\_samples\_split: minimum banyaknya sample/examples dalam sebuah node agar bisa di-split
  - min\_samples\_leaf: minimum banyaknya sample/examples agar sebuah node bisa menjadi leaf
  - max\_features: banyaknya fitur/atribut yang dipilih untuk calon split point
  - max\_leaf\_nodes: maksimum banyaknya leaf dalam sebuah tree
  - dll.

# Issues, Pros, & Cons of CART

## Sumber:

- Adila A. Krisnadhi, Slides Materi Pemelajaran Mesin, “CART and Random Forests”, Semester Genap 2020/2021
- Kevin P. Murphy, “Probabilistic Machine Learning: An Introduction”, MIT Press, 2021.

# Issues

## Missing Values

- Lakukan data imputation saat pre-processing training set
- ... atau gunakan *surrogate splits*
  - Ketika membangun tree, split yang disimpan tidak hanya split terbaik (pasangan terbaik variabel input beserta nilai threshold), tetapi juga split yang menggunakan variabel input berbeda sedemikian sehingga hasil splitting tersebut hampir sama bagusnya dengan “the best split”. Split alternatif ini adalah surrogate split dan bisa dipilih berdasarkan split yang mempunyai nilai cost mendekati “the best split”.
  - Bila suatu saat ada example yang seharusnya diuji dengan split terbaik, ternyata memiliki missing value pada nilai variabel input yang ditanyakan, maka surrogate split dapat digunakan sebagai alternatif.

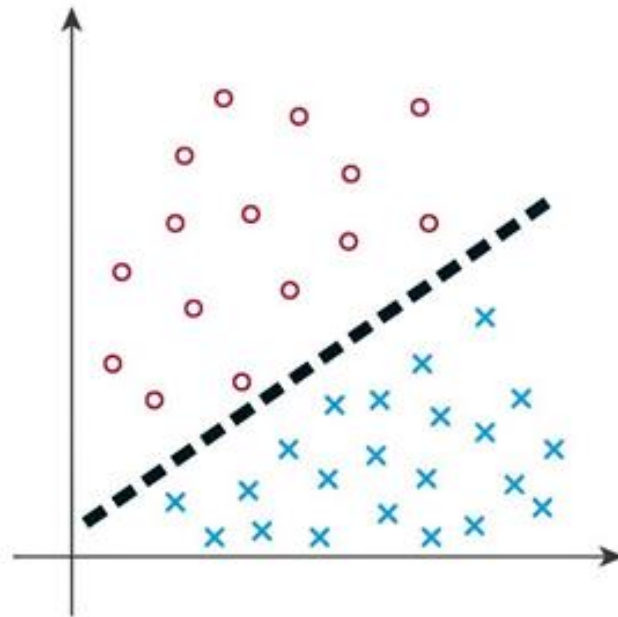
## Multi-way split & berhenti ketika nilai variabel output sama

- Dapat menyebabkan data fragmentation, yaitu keadaan sebuah sub-tree mengandung data yang terlalu sedikit, yang dapat menyebabkan overfitting.

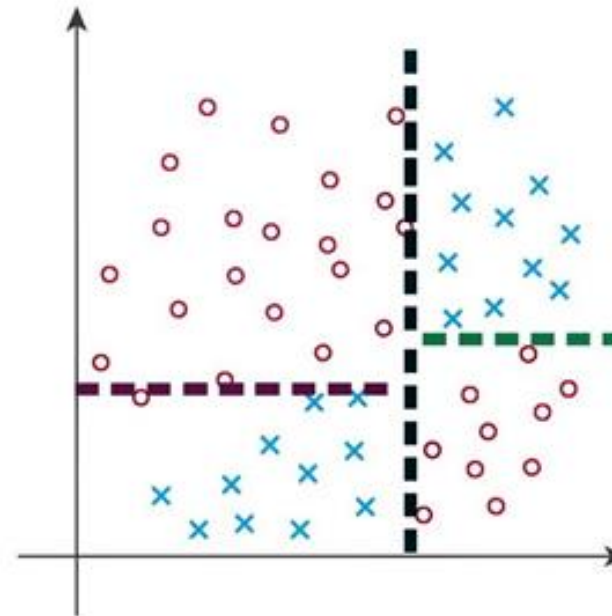
Overfitting?? Akan dibahas lebih lanjut di Evaluasi Model dan Bias-Variance Trade Off

# Issues

- Nonlinear model
- Sulit memisahkan data yang linearly separable dibandingkan dengan model linear lainnya

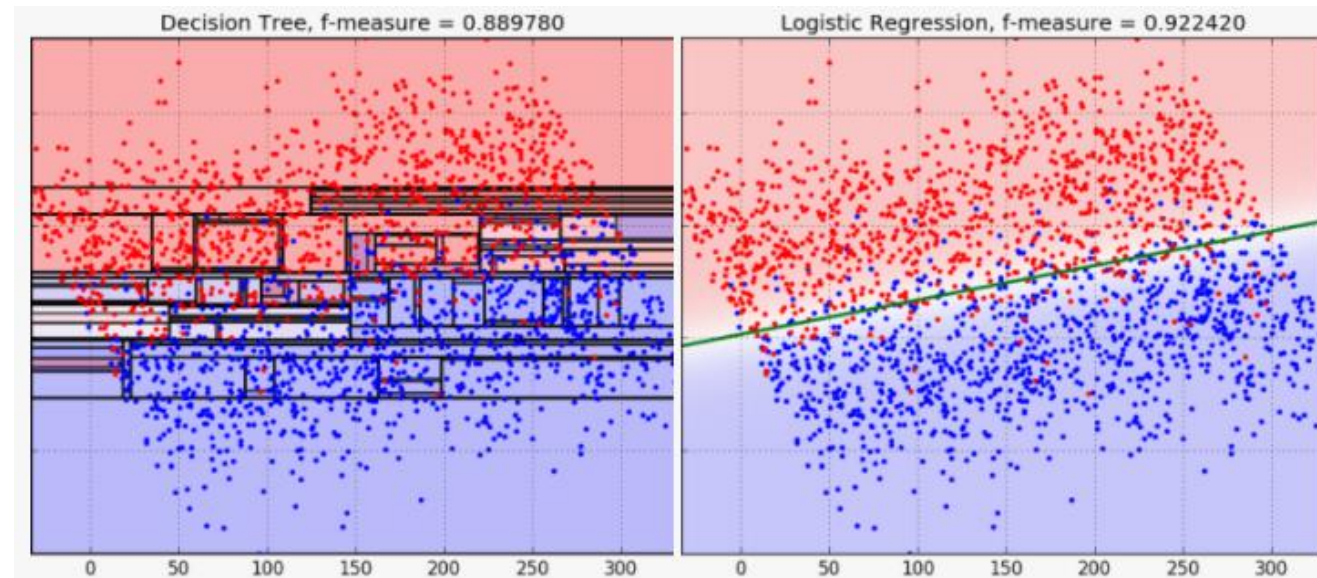
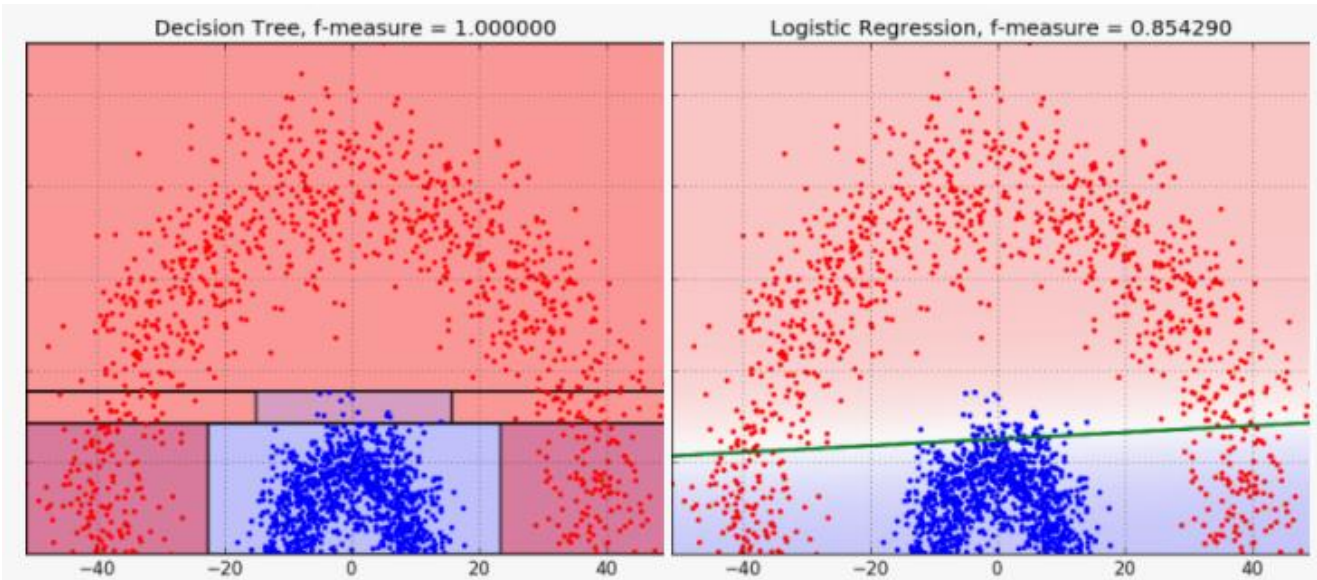


Linearly separable dataset



Linearly inseparable dataset

# Issues



Decision Tree vs Logistic Regression

Nonlinearly separable data vs Linearly separable data

<https://blog.bigml.com/2016/09/28/logistic-regression-versus-decision-trees/>



# Pros & Cons

Berdasarkan bagian sebelumnya, menurut Anda, apakah kelebihan dan kelemahan *decision tree* sebagai model prediksi?

(+) ....

(+) ....

(-) ....

(-) ....



# Pros & Cons

- (+) easy to interpret
- (+) easily handle mixed discrete and continuous inputs
- (+) insensitive to monotone transformation (because split points are based on ranking data points), hence no need to standardize data
- (+) perform automatic variable/feature selection
- (+) relatively robust to outliers
- (+) fast to fit, and scale well to large datasets
- (+) can handle missing input features (using some heuristics)
  
- (+/-) nonlinear classifier
  
- (-) do not predict very accurately compared to other kinds of models (due to its greedy nature)
- (-) unstable, small changes to input data can cause large effects on the structure of the tree (due to hierarchical nature of tree construction) as errors at the top influence the rest of the tree. (Decision trees are high-variance estimators)



FAKULTAS  
ILMU  
KOMPUTER

# TERIMA KASIH

Disclaimer: Figures and content can be originated from other sources on the Web. The purpose of this slide set is educational only.