Klasifikasi

Diadaptasi dari slide **Jiawei Han**
http://www.cs.uiuc.edu/~hanj/bk2/

Pengantar

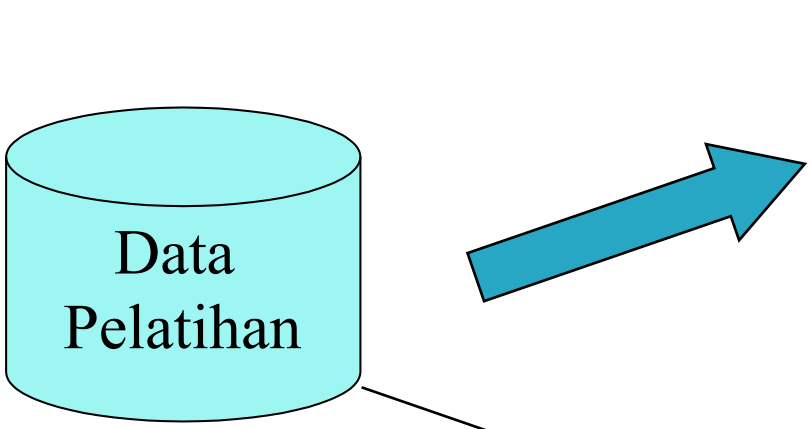
- Classification
  - Memprediksi kelas suatu item
  - Membuat model berdasarkan data pelatihan dan digunakan untuk mengklasifikasi data.
- Prediction
  - Memprediksi nilai yang belum diketahui
- Aplikasi
  - Persetujuan kredit
  - Diagnosis penyakit
  - Target marketing
  - Fraud detection

Contoh Kasus

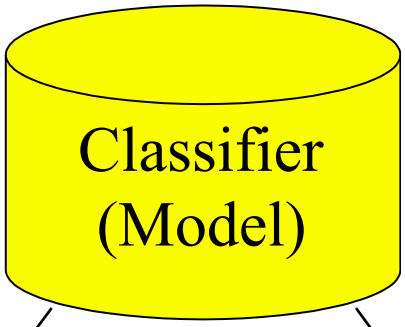
- Input: data mahasiswa
- Output: dua kelas (lulus_tepat_waktu dan lulus_terlambat)

Bagaimana kalau diberikan data input mahasiswa, sistem secara otomatis menentukan mhs tersebut akan lulus tepat waktu atau terlambat?

Pembuatan Model

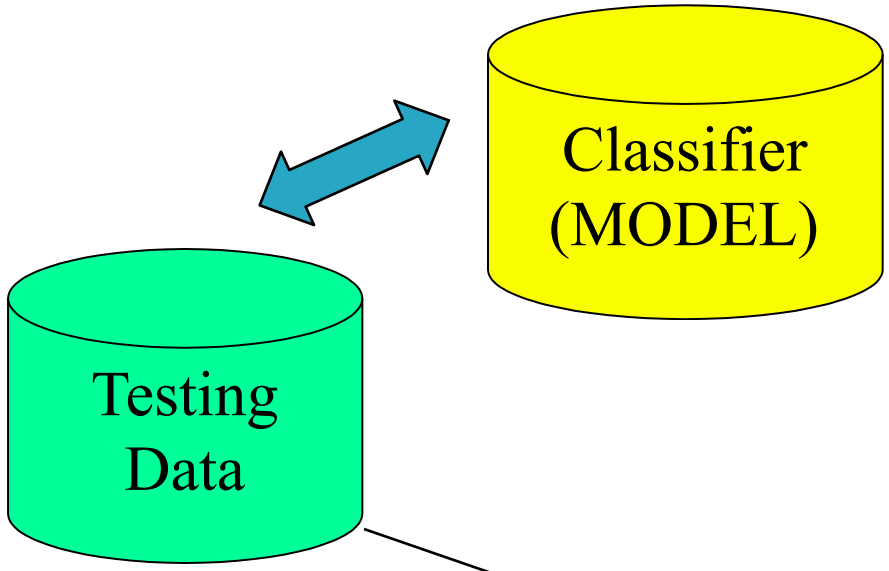


NAMA	IPK Sem 1	Matdas	tepat_waktu
Budi	3	A	yes
Wati	1.5	E	no
Badu	2	A	yes
Rudi	3.5	C	yes



IF IPK > 3
OR MATDAS =A
THEN tepat_waktu =
'yes'

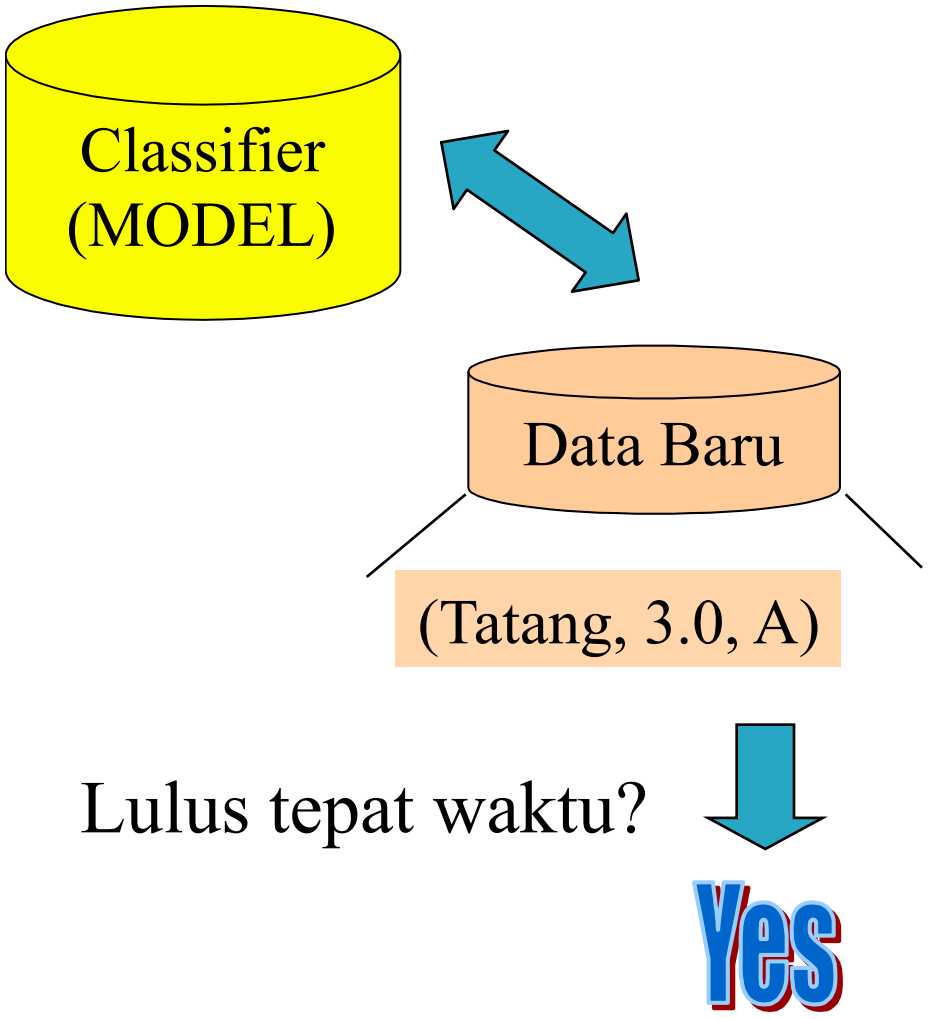
Proses Testing Model



NAMA	IPK_SEM1	MADAS	TEPAT_WAKTU
Akhmad	3.2	A	yes
Intan	3.3	B	no
Indah	2.3	C	yes
Ujang	1.7	E	no

Sejauh mana model tepat meramalkan?

Proses Klasifikasi



- Proses pembuatan model
  - Data latihan → Model Klasifikasi
- Proses testing model
  - Data testing → Apakah model sudah benar?
- Proses klasifikasi
  - Data yang tidak diketahui kelasnya → kelas data

Sebelum Klasifikasi

- Data cleaning
  - Preprocess data untuk mengurangi noise dan missing value
- Relevance analysis (feature selection)
  - Memilih atribut yang penting
  - Membuang atribut yang tidak terkait atau duplikasi.
- Data transformation
  - Generalize and/or normalize data

Evaluasi Metode Klasifikasi

- Akurasi
  - classifier accuracy: memprediksi label kelas
  - predictor accuracy: memprediksi nilai atribut
- kecepatan
  - Waktu untuk membuat model (training time)
  - Waktu untuk menggunakan model (classification/prediction time)
- Robustness: menangai noise dan missing value.
- Scalability: efisien untuk proses dengan DBMS
- Interpretability
  - Model mudah dimengerti

- Slide berikutnya… salah satu metode: decision tree

Decision Tree

- Diciptakan oleh Ross Quinlan
- ID3, C4.5, C5.0
- Model direpresentasikan dalam bentuk tree

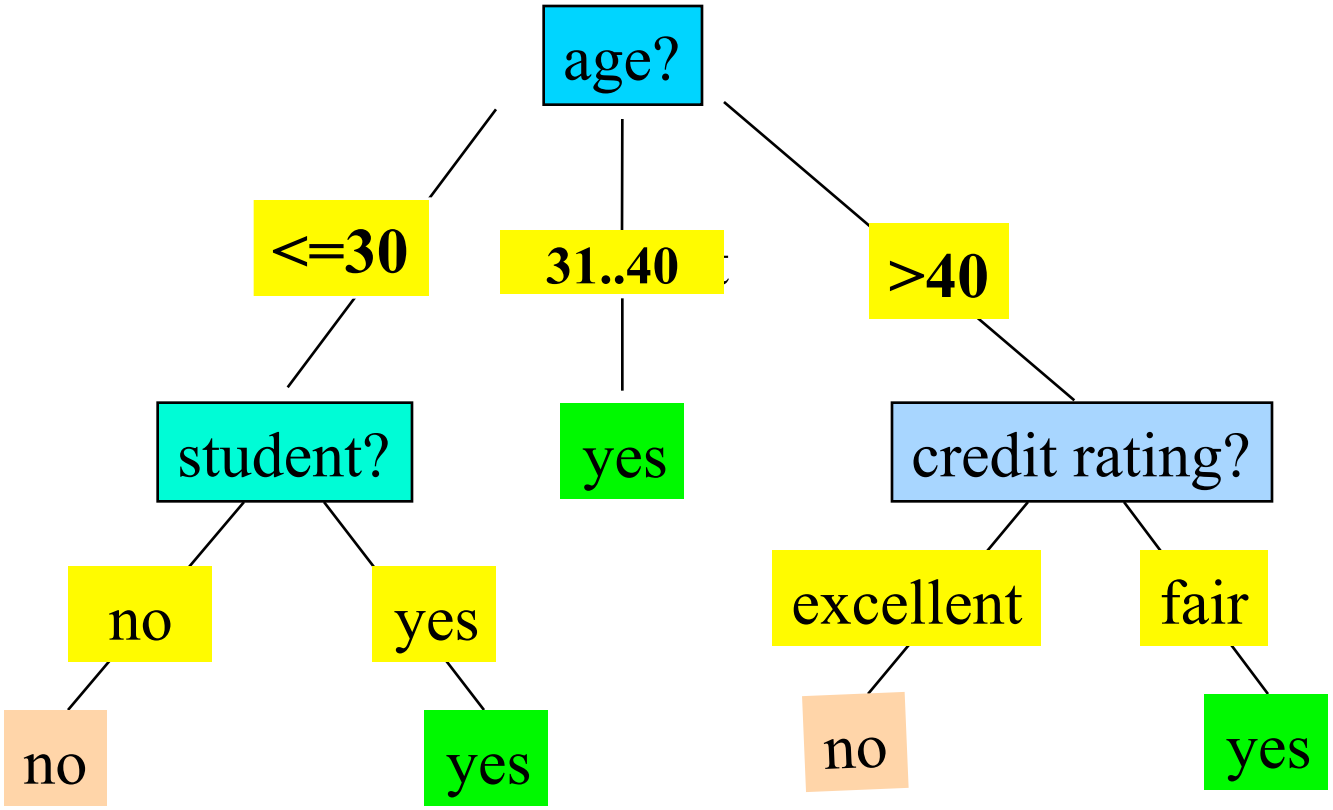
Decision Tree: Contoh Input (Data Latih)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Masalah

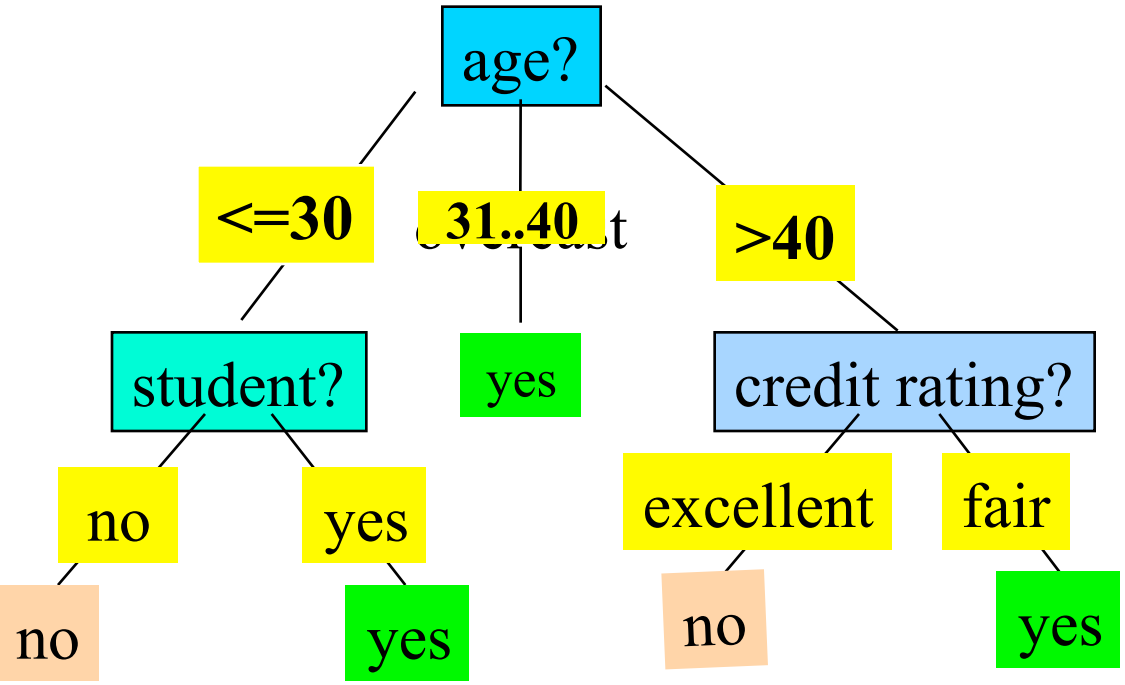
- Bagaimana dari data latih tersebut dapat diperoleh model yang bisa mengklasifikasikan secara otomatis?

Model: Decision Tree



Dari data latih, model ini dibangkitkan secara otomatis…

Tree Dapat Direpresentasikan sebagai Rule



((age<=30) and
(student))

OR

(age=31..40)

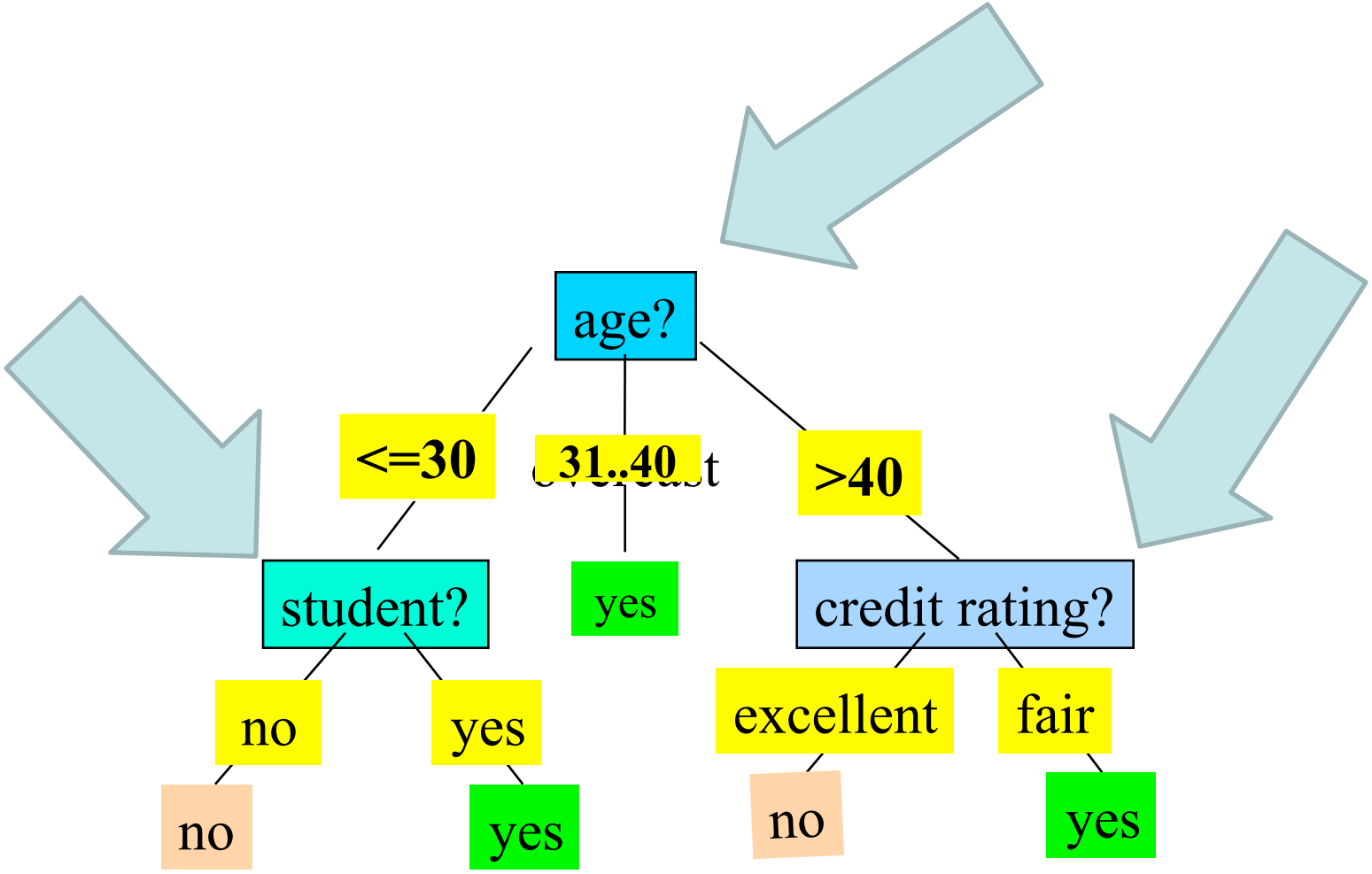
OR

(age>40) and
(credit_rating=fair)

THEN

BELI_PC=YES

Bagaimana cara pemilihan urutan atribut?



Cara Pemilihan Atribut

- Entrophy: Ukuran kemurnian, semakin murni, semakin homogen, semakin rendah nilainya.
- Information Gain: pengurangan entropy disebabkan oleh partisi berdasarkan suatu atribut.

  Semakin besar info gain = atribut itu semakin membuat homogen = semakin bagus

  Idenya → pilih atribut dengan info gain yg paling besar

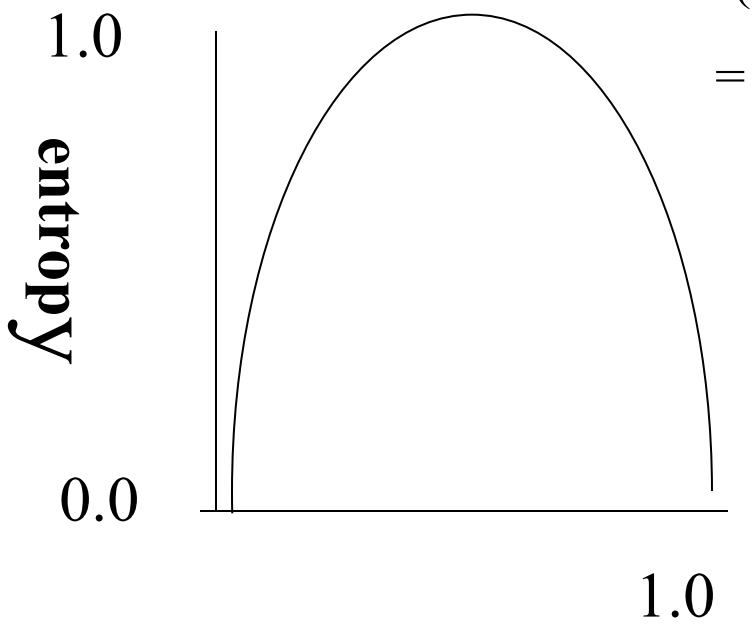
Entrophy untuk dua kelas: + dan -

$$Entropy(S) \equiv -p_{\oplus} \log_2 p - p_{\Theta} \log_2 p_{\Theta}$$

Entropy([9+,5-] ((9 positif, 5 neg)) =

$-(9/14) \log_2(9/14) - (5/14) \log_2(5/14)$

$= 0.940$



Entropy([9+,5-]) = 0.940

Entropy([7+,7-]) = 1

Entropy([14+,0]) = 0

Entroy([0+,14-]) = 0

Entrophy untuk kelas > 2

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Info (D) = Entrophy (D) (istilah dibuku J. HAN)

Information Gain

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

$$Gain(A) = Info(D) - Info_A(D)$$

Gain(A) seberapa besar entropy berkurang akibat atribut A. Makin besar makin bagus.

Contoh Pemilihan Atribut

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

$\frac{5}{14}I(2,3)$ berarti ada 5 dari 14 “age <=30” dgn 2 yes dan 3 no.

Gain (Age) = Info(D) – Info age (D)
=0.940 – 0.694 = 0.246

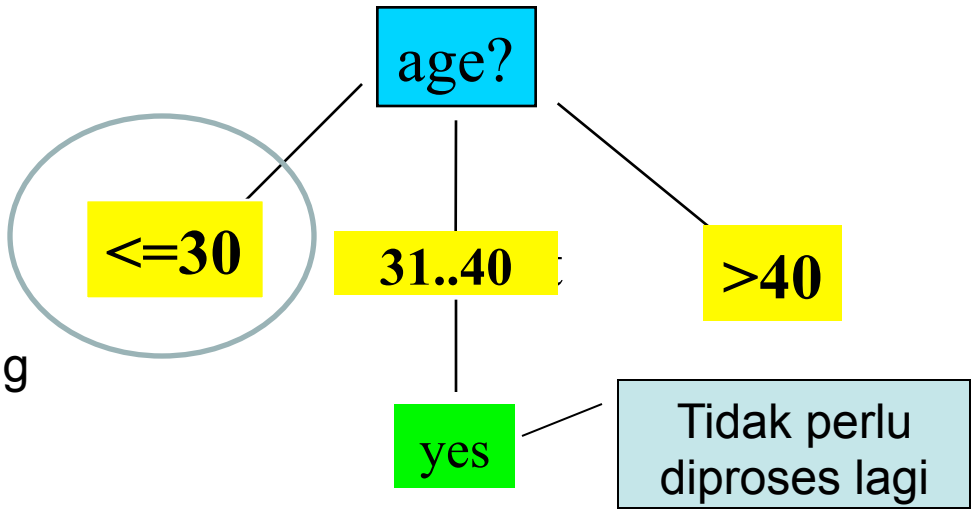
$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Pemilihan Atribut (lanj)

Gain (Age) = 0.246 ← yang terbesar, dipilih
Gain (income)=0.029
Gain(student)=0.151
Gain(credit_rating) =0.048

Setelah AGE, atribut apa selanjutnya?

Diproses untuk setiap cabang selama masih ada > 1 kelas



Selanjutnya... proses data yang <=30

Pemilihan Atribut (lanj)

Selanjutnya... proses data age<=30

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
<=30	medium	yes	excellent	yes

$$Info(D) = I(2,3) = -\frac{2}{5}\log_2(\frac{2}{5}) - \frac{3}{5}\log_2(\frac{3}{5}) = 0.97$$

Gain(age) tidak perlu dihitung lagi, hitung gain(student), gain(credit_rating)

$$Info_{student}(D) = \frac{3}{5}I(0,3) + \frac{2}{5}I(2,0) = 0$$

Gain (student) = Info(D) – Info$_{student}$(D)
=0.97 – 0 = 0.97

Pemilihan Atribut (lanj)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
<=30	medium	yes	excellent	yes

hitung gain(credit_rating)

$$Info(D) = I(2,3) = -\frac{2}{5}\log_2(\frac{2}{5}) - \frac{3}{5}\log_2(\frac{3}{5}) = 0.97$$

$$Info_{credit\_rating}(D) = \frac{3}{5}I(1,2) + \frac{2}{5}I(1,1) = 0.95$$

Gain (credit_rating) = Info(D) – Info$_{student}$(D)
=0.97 – 0.95 = 0.02

$$Info_{income}(D) = \frac{2}{5}I(0,2) + \frac{2}{5}I(1,1) + \frac{1}{5}I(1,0) = 0.4$$

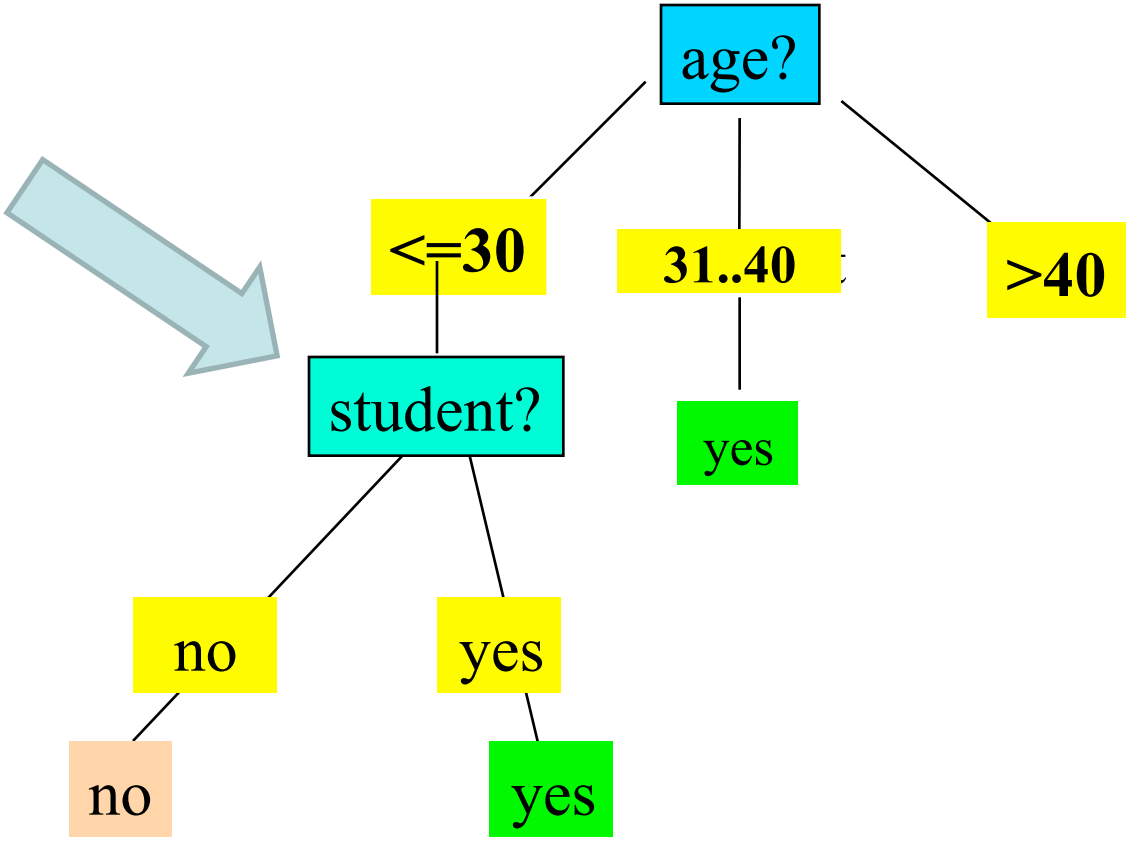
Pilihan Atribut (lanj)

Bandingkan semua gain, ambil yang paling besar

Gain (studet) = 0.97
Gain (credit_rating = 0.02
Gain (income) = 0.4

Paling besar
student

Pemilhan Atribut (lanj)



Latihan

No	Kelas	Kulit Buah	Warna	Ukuran	Bau
1	Aman	Kasar	Coklat	Besar	keras
2	Aman	Kasar	Hijau	Besar	keras
3	Berbahaya	Halus	Merah	Besar	Lunak
4	Aman	Kasar	Hijau	Besar	Lunak
5	Aman	Kasar	Merah	Kecil	Keras
6	Aman	Halus	Merah	Kecil	Keras
7	Aman	Halus	Coklat	Kecil	Keras
8	Berbahaya	Kasar	Hijau	Kecil	Lunak
9	Berbahaya	Halus	Hijau	Kecil	Keras
10	Aman	Kasar	Merah	Besar	Keras
11	Aman	Halus	Coklat	Besar	Lunak
12	Berbahaya	Halus	Hijau	Kecil	Keras
13	Aman	Kasar	Merah	Kecil	Lunak
14	Berbahaya	Halus	Merah	Besar	Keras
15	Aman	Halus	Merah	Kecil	Keras
16	Berbahaya	Kasar	Hijau	Kecil	Keras

Mengapa Decision Tree?

- Mudah diimplementasikan
- Hipotesis yang dihasilkan mudah dipahami
- Efisien

Decision Tree Cocok untuk Masalah:

- Data dalam bentuk atribut-nilai. Kondisi ideal adalah jika isi nilai jumlahnya sedikit. Misalnya: “panas”, “sedang”, “dingin”.
- Output diskrit.
- Training data dapat tidak lengkap

Masalah DT

- Overfitting: terlalu mengikuti training data
  - Terlalu banyak cabang, merefleksikan anomali akibat noise atau outlier.
  - Akurasi rendah untuk data baru
- Dua pendekatan untuk menghindari overfitting
  - Prepruning: Hentikan pembuatan tree di awal. Tidak mensplit node jika goodness measure dibawah threshold.
    - Sulit untuk menentukan threshold
  - Postpruning: Buang cabang setelah tree jadi
    - Menggunakan data yang berbeda dengan training untuk menentukan pruned tree yang terbaik.

Bayesian Classification

- P(H | X) Kemungkinan H benar jika X. X adalah kumpulah **atribut.**
- P(H) Kemungkinan H di data, independen terhadap X
- P (“Single” | “muka sayu”, “baju berantakan”, “jalan sendiri”) → nilainya besar
- P (“Non Single” | “muka ceria”, “baju rapi”, “jalan selalu berdua”) → nilainya besar
- P (“Single”) = jumlah single / jumlah mahasiswa

- P(H | X) → posterior
- P(H) → a priori
- P (X | H) probabilitas X, jika kita ketahui bahwa H benar → data training
- Kegiatan klasifikasi: kegiatan mencari P (H | X) yang paling maksimal
- Teorema Bayes:

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$$

Klasifikasi

X = (“muka cerah”, “jalan sendiri”, “baju rapi”)

Kelasnya Single atau Non Single?

Cari P(H|X) yang paling besar:

(“Single” | “muka cerah”, “jalan sendiri”, “baju rapi”)

Atau

(“**Non** Single” | “muka cerah”, “jalan sendiri”, “baju rapi”)

Harus memaksimalkan (C_i: kelas ke i)

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

Karena P(X) konstan untuk setiap C*i* maka bisa ditulis, pencarian max untuk:

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

Naïve Bayes Classifier

- Penyederhanaan masalah: Tidak ada kaitan antar atribut “jalan sendiri” tidak terakait dengan “muka sayu”

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times ... \times P(x_n|C_i)$$

X_1: atribut ke-1 (“jalan sendiri”)

X_n: atribut ke-n

Naïve Bayes

- Jika bentuknya kategori ,

  $P(x_k|C_i)$ = jumlah kelas C_i yang memiliki x_k dibagi $|C_i|$ (jumlah anggota kelas C_i di data contoh)
- Jika bentuknya continous dapat menggunakan distribusi gaussian

Contoh Naïve Bayes

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Contoh Naïve Bayes

P(Ci):
P(buys_computer = “yes”) = 9/14 = 0.643
P(buys_computer = “no”) = 5/14= 0.357

Training: Hitung P(X|Ci) untuk setiap kelas
P(age = “<=30” | buys_computer = “yes”) = 2/9 = 0.222
P(age = “<= 30” | buys_computer = “no”) = 3/5 = 0.6
P(income = “medium” | buys_computer = “yes”) = 4/9 = 0.444
P(income = “medium” | buys_computer = “no”) = 2/5 = 0.4
P(student = “yes” | buys_computer = “yes) = 6/9 = 0.667
P(student = “yes” | buys_computer = “no”) = 1/5 = 0.2
P(credit_rating = “fair” | buys_computer = “yes”) = 6/9 = 0.667
P(credit_rating = “fair” | buys_computer = “no”) = 2/5 = 0.4

Klasifikasi: X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

P(X|Ci) : P(X|buys_computer = “yes”) = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = “no”) = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

P(X|Ci)*P(Ci) :
P(X|buys_computer = “yes”) * P(buys_computer = “yes”) = 0.028 ← P(X| buys_computer = “no”) * P(buys_computer = “no”) = 0.007

Pro, Cons Naïve Bayes

- Keuntungan
  - Mudah untuk dibuat
  - Hasil bagus
- Kerugian
  - Asumsi independence antar atribut membuat akurasi berkurang (karena biasanya ada keterkaitan)

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
  - Supervision: Data pelatihan mengandung label kelas.
  - Data diklasifikasikan menggunakan model.
- Unsupervised learning (clustering)
  - Data pelatihan tidak mengandung label kelas
  - Mencari kelas atau cluster di dalam data.
  - Akan dijelaskan terpisah