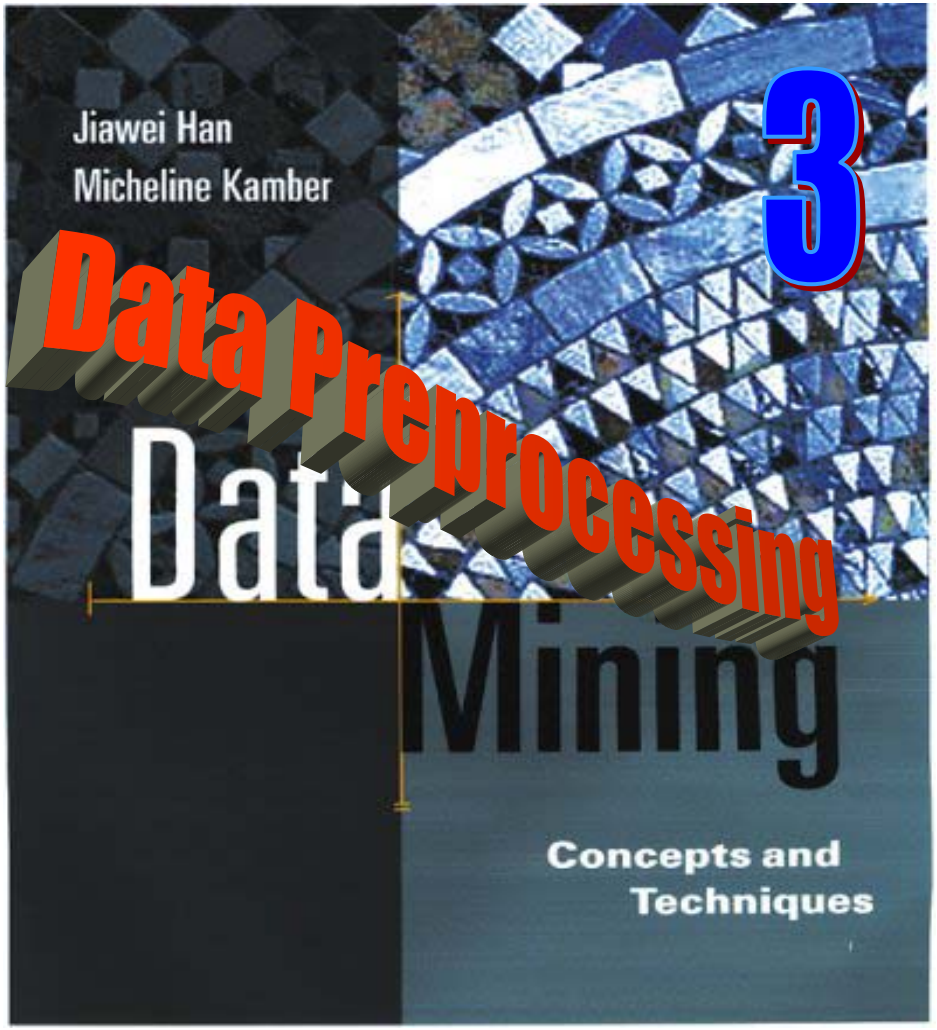
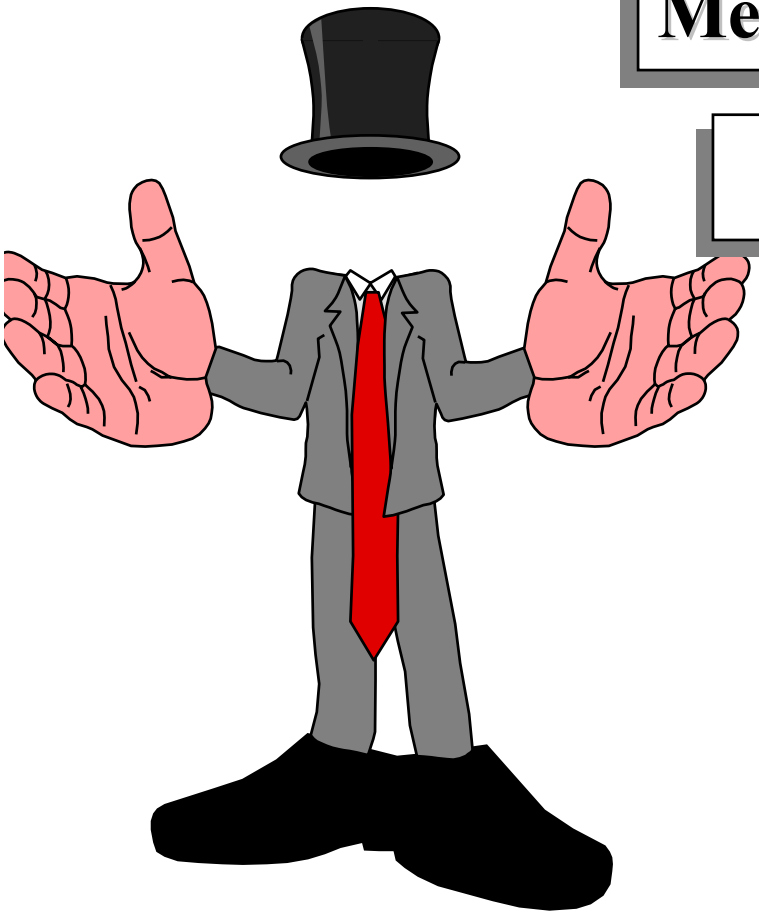
# Konsep dan Teknik Data Mining



# Data Preprocessing



- Mengapa data di proses awal?
- Pembersihan data
- Integrasi dan transformasi data
- Reduksi data
- Diskritisasi dan pembuatan konsep hierarki

# Mengapa Data Diproses Awal?

- Data dalam dunia nyata kotor
  - Tak-lengkap: nilai-nilai atribut kurang, atribut tertentu yang dipentingkan tidak disertakan, atau hanya memuat data agregasi
    - Misal, pekerjaan=“”
  - Noisy: memuat error atau memuat outliers (data yang secara nyata berbeda dengan data-data yang lain)
    - Misal, Salary=“-10”

# Mengapa Data Diproses Awal?

- Tak-konsisten: memuat perbedaan dalam kode atau nama
  - Misal, Age="42" Birthday="03/07/1997"
  - Misal, rating sebelumnya "1,2,3", sekarang rating "A, B, C"
  - Misal, perbedaan antara duplikasi record
- Data yang lebih baik akan menghasilkan data mining yang lebih baik
- Data preprocessing membantu didalam memperbaiki presisi dan kinerja data mining dan mencegah kesalahan didalam data mining.

# Mengapa Data Kotor?

- Ketaklengkapan data datang dari
  - Nilai data tidak tersedia saat dikumpulkan
  - Perbedaan pertimbangan waktu antara saat data dikumpulkan dan saat data dianalisa.
  - Masalah manusia, hardware, dan software
- Noisy data datang dari proses data
  - Pengumpulan
  - Pemasukan (entry)
  - Transmisi

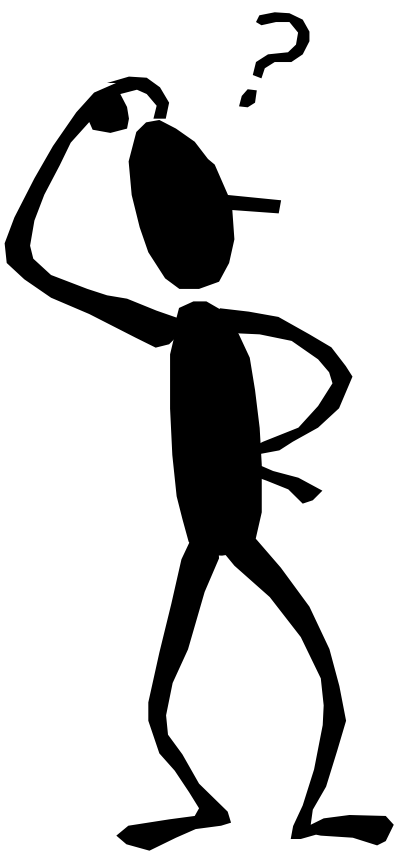
# Mengapa Data Kotor?

- Ketak-konsistenan data datang dari
  - Sumber data yang berbeda
  - Pelanggaran kebergantungan fungsional

# Mengapa Pemrosesan Awal Data Penting?

- Kualitas data tidak ada, kualitas hasil mining tidak ada!
  - Kualitas keputusan harus didasarkan kepada kualitas data
    - Misal, duplikasi data atau data hilang bisa menyebabkan ketidak-benaran atau bahkan statistik yang menyesatkan.
  - Data warehouse memerlukan kualitas integrasi data yang konsisten
- Ekstraksi data, pembersihan, dan transformasi merupakan kerja utama dari pembuatan suatu data warehouse. — Bill Inmon

# Pengukuran Kualitas Data Multidimesi



- Kualitas data dapat diakses dalam bentuk:
  - Akurasi
  - Kelengkapan
  - Konsistensi
  - Ketepatan waktu
  - Kepercayaan
  - Nilai tambah
  - Penafsiran
  - Kemudahan diakses
- Kategori luas:
  - Hakekat, kontekstual, bisa direpresentasikan, dan mudah diakses

# Tugas Utama Pemrosesan Awal Data

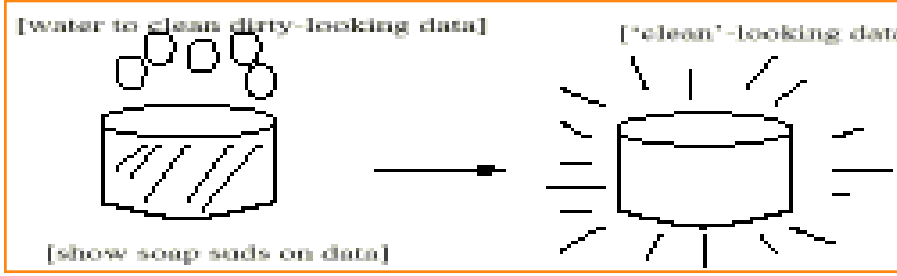
- Pembersihan data (data yang kotor)
  - Mengisi nilai-nilai yang hilang, menghaluskan noisy data, mengenali atau menghilangkan outlier, dan memecahkan ketak-konsistenan
- Integrasi data (data heterogen)
  - Integrasi banyak database, banyak kubus data, atau banyak file
- Transformasi data (data detail)
  - Normalisasi dan agregasi

# Tugas Utama Pemrosesan Awal Data

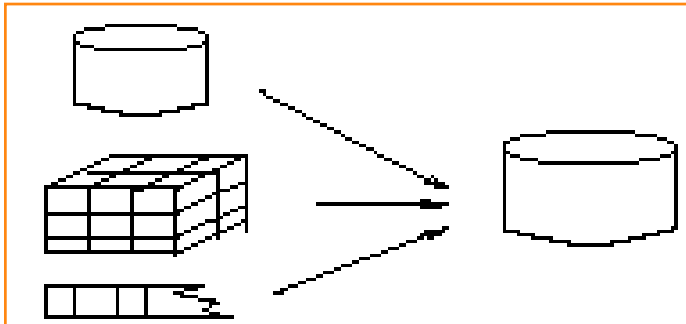
- Reduksi data (jumlah data yang besar)
  - Mendapatkan representasi yang direduksi dalam volume tetapi menghasilkan hasil analitikal yang sama atau mirip
- Diskritisasi data (kesinambungan atribut)
  - Bagian dari reduksi data tetapi dengan kepentingan khusus, terutama data numerik

# Bentuk-Bentuk Dari Pemrosesan Awal Data

**Pembersihan Data**



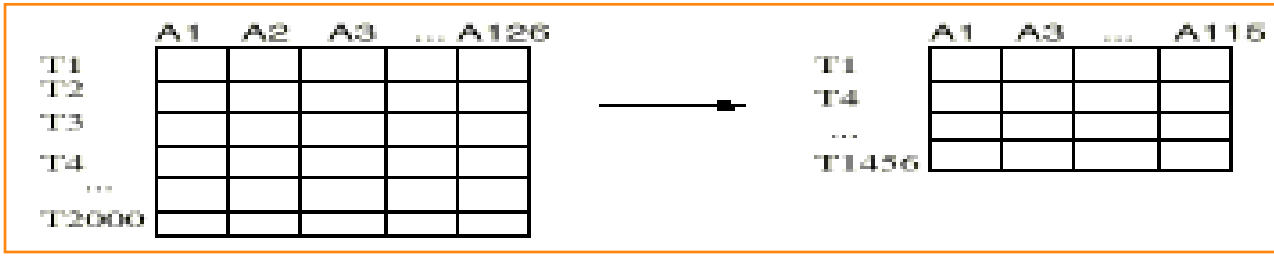

**Integrasi Data**



**Transformasi Data**

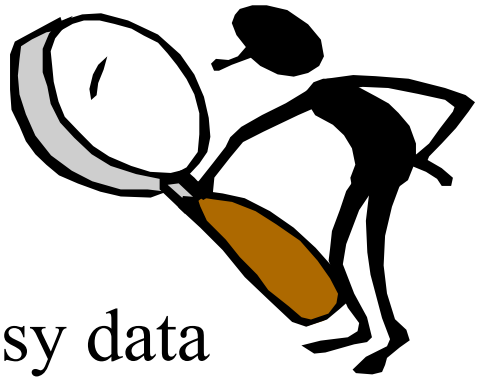
-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

**Reduksi Data**



# Pembersihan Data

- Kepentingan
  - "Pembersihan data adalah salah satu dari 3 problem terbesar dalam data warehousing"—Ralph Kimball
  - "Pembersihan data adalah problem nomor 1 dalam data warehousing"—DCI survey
- Tugas pembersihan data
  - Mengisi nilai-nilai yang hilang
  - Mengenali outliers dan menghaluskan noisy data
  - Memecahkan redundansi yang disebabkan oleh integrasi data

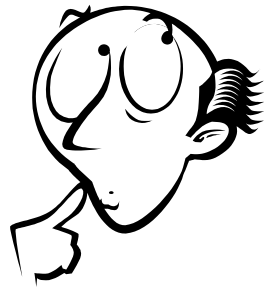


# Pembersihan Data

- Memperbaiki ketak-konsitenan data, US=USA?
  - Menggunakan rujukan eksternal
  - Mendeteksi pelanggaran kendala
    - Misal, kebergantungan fungsional

# Data Hilang

- Data tidak selalu tersedia
  - Misal, banyak tuple atau record tidak memiliki nilai yang tercatat untuk beberapa atribut, seperti customer income dalam data sales
- Hilangnya data bisa karena
  - Kegagalan pemakaian peralatan
  - Ketak-konsistenan dengan data tercatat lainnya dan karenanya dihapus
  - Data tidak dimasukkan karena salah pengertian
  - Data tertentu bisa tidak dipandang penting pada saat entry



# Data Hilang

  - Tidak mencatat history atau tidak mencatat perubahan data
- Kehilangan data perlu disimpulkan

# Bagaimana Menangani Data Hilang?

- Mengabaikan tuple atau record: mudah tetapi tidak efektif, dan merupakan metoda terakhir
  - Biasanya dilakukan saat label kelas hilang
  - Tidak efektif bila persentasi dari nilai-nilai yang hilang per atribut sungguh-sungguh bervariasi.
- Mengisi nilai-nilai yang hilang secara manual:
  - Paling baik
  - Membosankan
  - Paling mahal biayanya
  - Tak mungkin dilakukan dalam banyak hal!

# Bagaimana Menangani Data Hilang?

- Mengisi nilai-nilai yang hilang secara otomatis menggunakan:
  - Suatu konstanta global: misal, “unknown”, “Null”, atau suatu kelas baru?!
    - Suatu pola yang memuat “unknown” atau “Null” adalah buruk
  - Gunakan rata-rata atribut
  - Pengempisan data ke mean/median
  - Rata-rata atribut untuk seluruh sampel yang masuk kedalam kelas yang sama
    - Lebih cerdas, dan suatu metoda yang baik

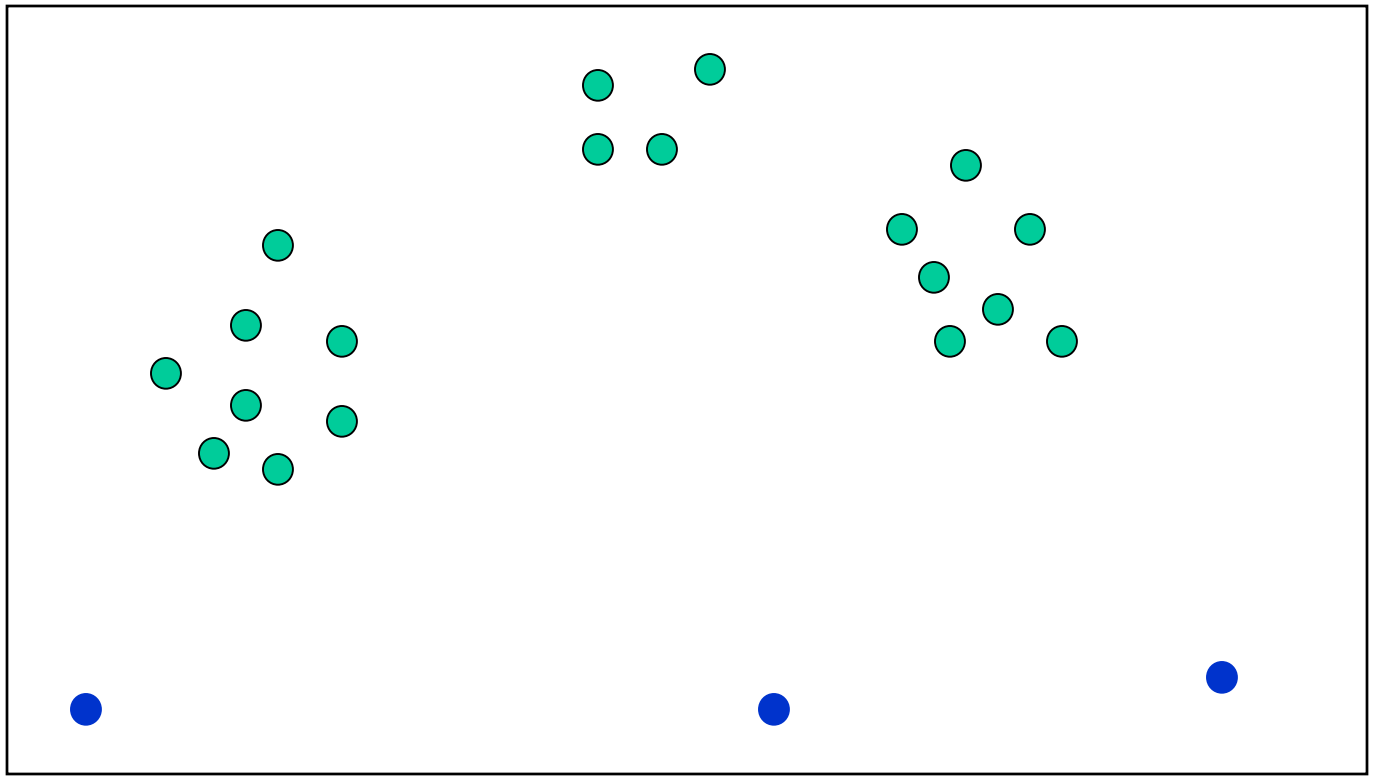
# Bagaimana Menangani Data Hilang?

- Nilai yang paling mungkin: berbasis inferensi seperti regresi, rumus bayesian, atau pohon keputusan
  - Klasifikasi untuk mendapatkan nilai yang paling mungkin
  - Suatu metoda yang baik dengan beberapa overhead
- Menggunakan suatu nilai untuk mengisi nilai yang hilang bisa membiaskan data, nilai bisa salah
- Nilai yang paling mungkin adalah yang terbaik
- Gunakan informasi yang paling banyak dari data yang ada untuk memprediksi

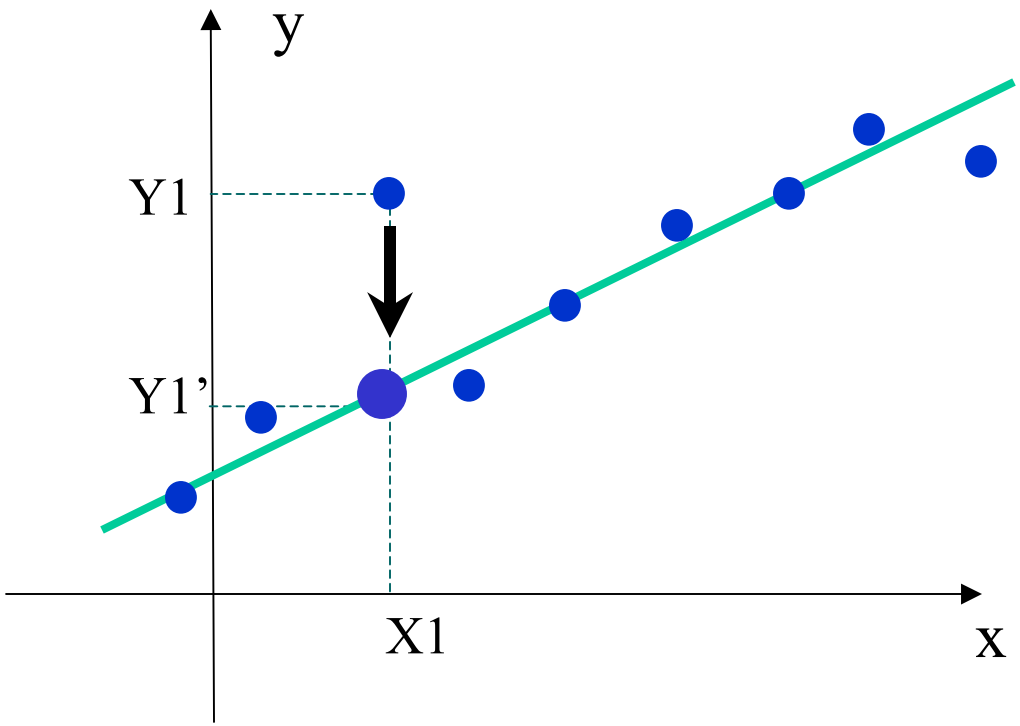
# Noisy Data

- Noise: error acak atau variansi dalam suatu variabel terukur
- Nilai-nilai atribut tak benar mungkin karena
  - Kegagalan instrumen pengumpulan data
  - Problem pemasukan data
  - Problem transmisi data
  - Keterbatasan teknologi
  - Ketak-konsistenan dalam konvensi penamaan
- Problem data lainnya yang memerlukan pembersihan data
  - Duplikasi record
  - Data tak lengkap
  - Data tidak konsisten

# Noisy Data: Menghilangkan Outlier



# Noisy Data: Penghalusan



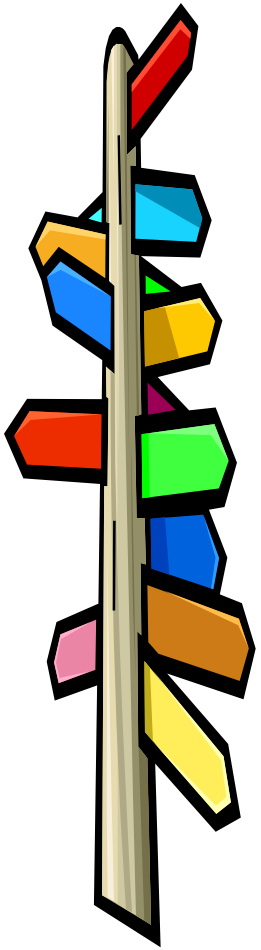
# Bagaimana Menangani Noisy Data?

- Metoda Binning:
  - Pertama urutkan data dan partisi kedalam (kedalaman yang sama) bin-bin
  - Kemudian noisy data itu bisa dihaluskan dengan rata-rata bin, median bin, atau batas bin.
- Clustering
  - Medeteksi dan membuang outliers
- Inspeksi kombinasi komputer dan manusia
  - Mendeteksi nilai-nilai yang mencurigakan dan memeriksa dengan manusia(misal, berurusan dengan outlier yang mungkin)

# Bagaimana Menangani Noisy Data?

- Regresi
  - Menghaluskan dengan memasukkan data kedalam fungsi regresi

# Metoda Binning: Diskritisasi Sederhana



- Partisi lebar yang sama (jarak):
  - Membagi range kedalam N interval dengan ukuran yang sama: grid seragam
  - Jika $A$ dan $B$ masing-masing adalah nilai terendah dan tertinggi dari atribut, lebar interval akan menjadi : $W = (B - A)/N$.
  - Kebanyakan langsung, tetapi outlier mendominasi presentasi
  - Data Outlier dan menyimpang tidak ditangani dengan baik.

# Metoda Binning: Diskritisasi Sederhana

- Partisi kedalaman sama (frekuensi):
  - Membagi range kedalam *N* interval, masing-masing memuat jumlah sampel yang hampir sama
  - Penskalaan data yang baik
  - Penanganan atribut yang bersifat kategori bisa rumit.

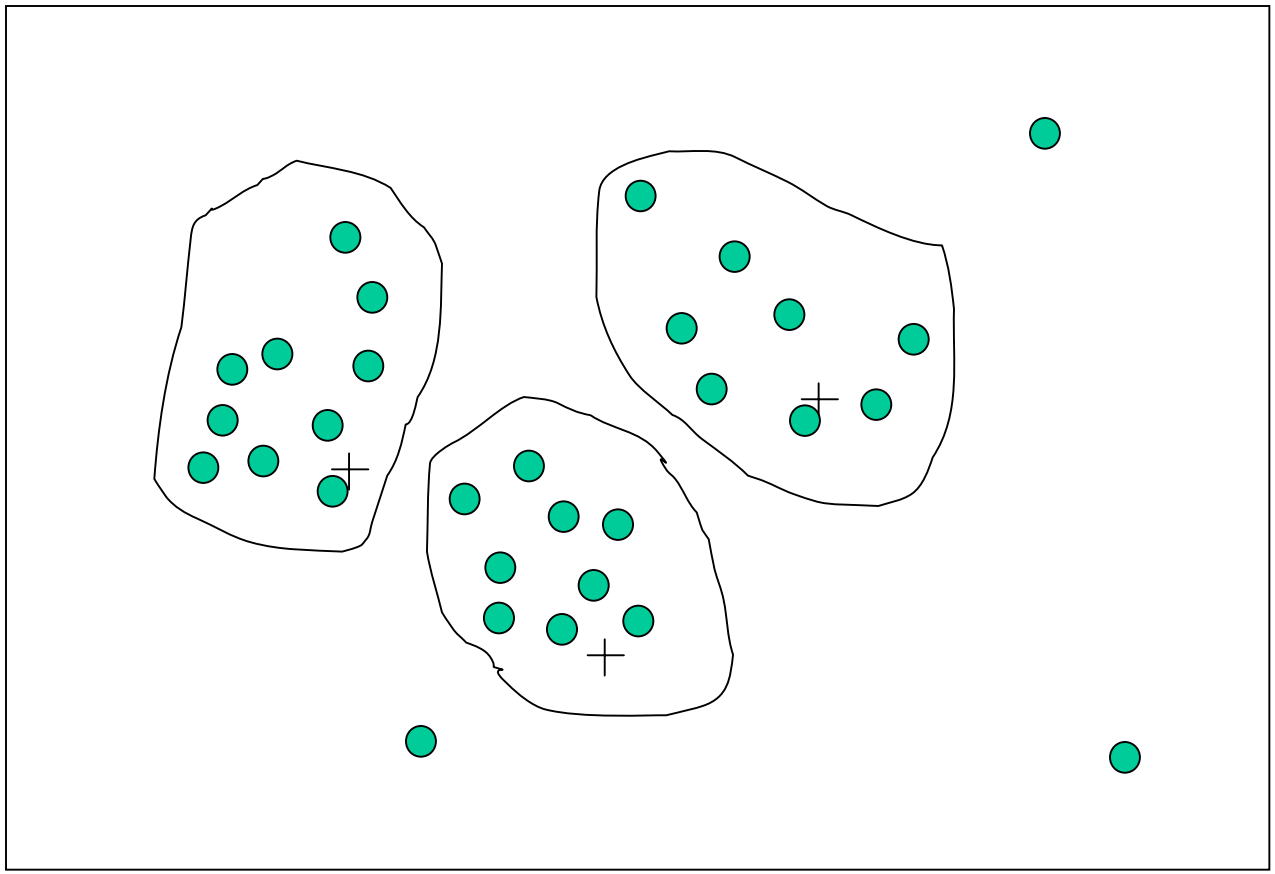
# Metoda Binning
# Untuk Penghalusan Data

- Data terurut untuk harga (dalam dollar): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- Partisi kedalam bin dengan kedalaman yang sama (misal, dalam bin-3):
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
- Haluskan dengan rata-rata bins:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29

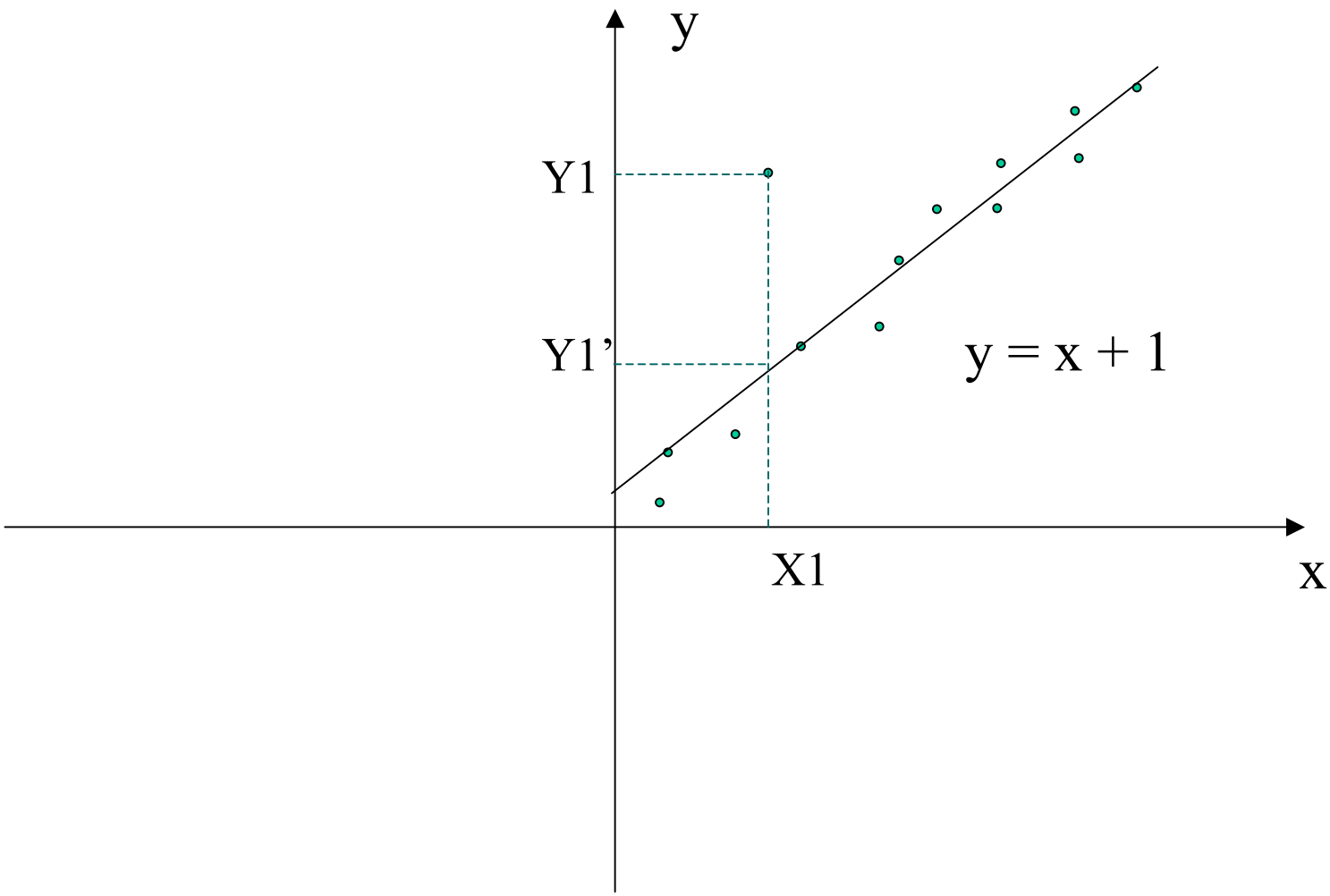
# Metoda Binning
# Untuk Penghalusan Data

- Penghalusan dengan batas bin:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

# Analisis Cluster



# Regresi



# Inspeksi Komputer dan Manusia Penghalusan

- Inspeksi kombinasi komputer dan manusia
  - Suatu ambang yang diberikan user
  - Komputer mendeteksi seluruh potensi outlier yang dikaitkan dengan ambang
  - Manusia menentukan outlier sesungguhnya

# Integrasi Data

- Integrasi data:
  - Mengkombinasikan data dari banyak sumber kedalam suatu simpanan terpadu
- Integrasi skema
  - Mengintegrasikan metadata dari sumber-sumber berbeda
  - Problem identifikasi entitas: mengenali entitas dunia nyata dari banyak sumber-sumber data, misal A.cust-id ≡ B.cust-#
- Pendeteksian dan pemecahan konflik nilai data
  - Untuk entitas dunia nyata yang sama, nilai-nilai atribut dari sumber-sumber berbeda adalah berbeda
  - Alasan yang mungkin: representasi berbeda, skala berbeda, misal berat bisa dalam pound atau kilogram

# Integrasi Data

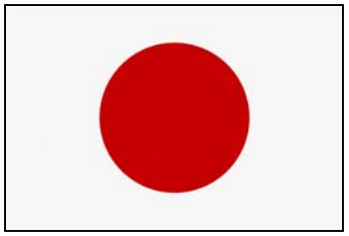
- Problem: integrasi skema heterogen
- Nama-nama tribut berbeda

| cid | name | byear |
|---|---|---|
| 1 | Jones | 1960 |
| 2 | Smith | 1974 |
| 3 | Smith | 1950 |

| Customer-ID | state |
|---|---|
| 1 | NY |
| 2 | CA |
| 3 | NY |

- Unit berbeda: Sales dalam $, sales dalam Yen, sales dalam DM





# Integrasi Data

- Problem: integrasi skema heterogen
- Skala berbeda: Sales dalam dollar versus sales dalam sen dollar

- Atribut turunan: Annual salary versus monthly salary

| cid | monthlySalary |
|---|---|
| 1 | 5000 |
| 2 | 2400 |
| 3 | 3000 |

| cid | Salary |
|---|---|
| 6 | 50,000 |
| 7 | 100,000 |
| 8 | 40,000 |

# Integrasi Data

- Problem: ketak-konsistenan karena redundansi
- Customer dengan customer-id 150 punya 3 anak dalam relation1 dan 4 anak dalam relation2

| cid | numChildren |
|---|---|
| 1 | 3 |

| cid | numChildren |
|---|---|
| 1 | 4 |

- Komputasi annual salary dari monthly salary dalam relation1 tak cocok dengan atribut “annual-salary” dalam relation2

| cid | monthlySalary |
|---|---|
| 1 | 5000 |
| 2 | 6000 |

| cid | Salary |
|---|---|
| 1 | 60,000 |
| 2 | 80,000 |

# Penanganan Redundansi Dalam Integrasi Data

- Data redundan sering terjadi saat integrasi dari banyak database
  - Atribut yang sama bisa memiliki nama berbeda dalam database berbeda
  - Atribut yang satu bisa merupakan suatu atribut “turunan” dalam tabel lainnya, misal, annual revenue
- Data redundan mungkin bisa dideteksi dengan analisis korelasi
- Integrasi data hati-hati dari banyak sumber bisa membantu mengurangi/mencegah redundansi dan ketak-konsistenan dan memperbaiki kecepatan dan kualitas mining

# Penanganan Redundansi Dalam Integrasi Data

- Suatu atribut adalah redundan jika atribut tersebut bisa diperoleh dari atribut lainnya
- Analisis korelasi $$r_{A,B} = \frac{\sum (A - \overline{A})(B - \overline{B})}{(n-1)\sigma_A \sigma_B}$$
- Rata-rata A adalah $$\overline{A} = \frac{\sum A}{n}$$
- Deviasi standard A adalah $$\sigma_A = \sqrt{\frac{\sum (A - \overline{A})^2}{n-1}}$$
- $R_{A,B} = 0$: A dan B saling bebas
- $R_{A,B} > 0$: A dan B berkorelasi positip $A\uparrow \leftrightarrow B\uparrow$
- $R_{A,B} < 0$: A dan B berkorelasi negatif $A\downarrow \leftrightarrow B\uparrow$

# Transformasi Data

- Penghalusan: menghilangkan noise dari data
- Agregasi: ringkasan, konstruksi kubus data
- Generalisasi: konsep hierarchy climbing
- Normalisasi: diskalakan agar jatuh didalam suatu range kecil yang tertentu
  - Normalisasi min-max
  - Normalisasi z-score
  - Normalisasi dengan penskalaan desimal
- Konstruksi atribut/fitur
  - Atribut-atribut baru dibangun dari atribut-atribut yang ada

# Transformasi Data: Normalisasi

- Normalisasi min-max

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- Normalisasi z-score (saat Min, Max tak diketahui)

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- Normalisasi dengan penskalaan desimal

$$v' = \frac{v}{10^j}$$ dimana $j$ adalah integer terkecil sehingga Max(| $v'$|)<1

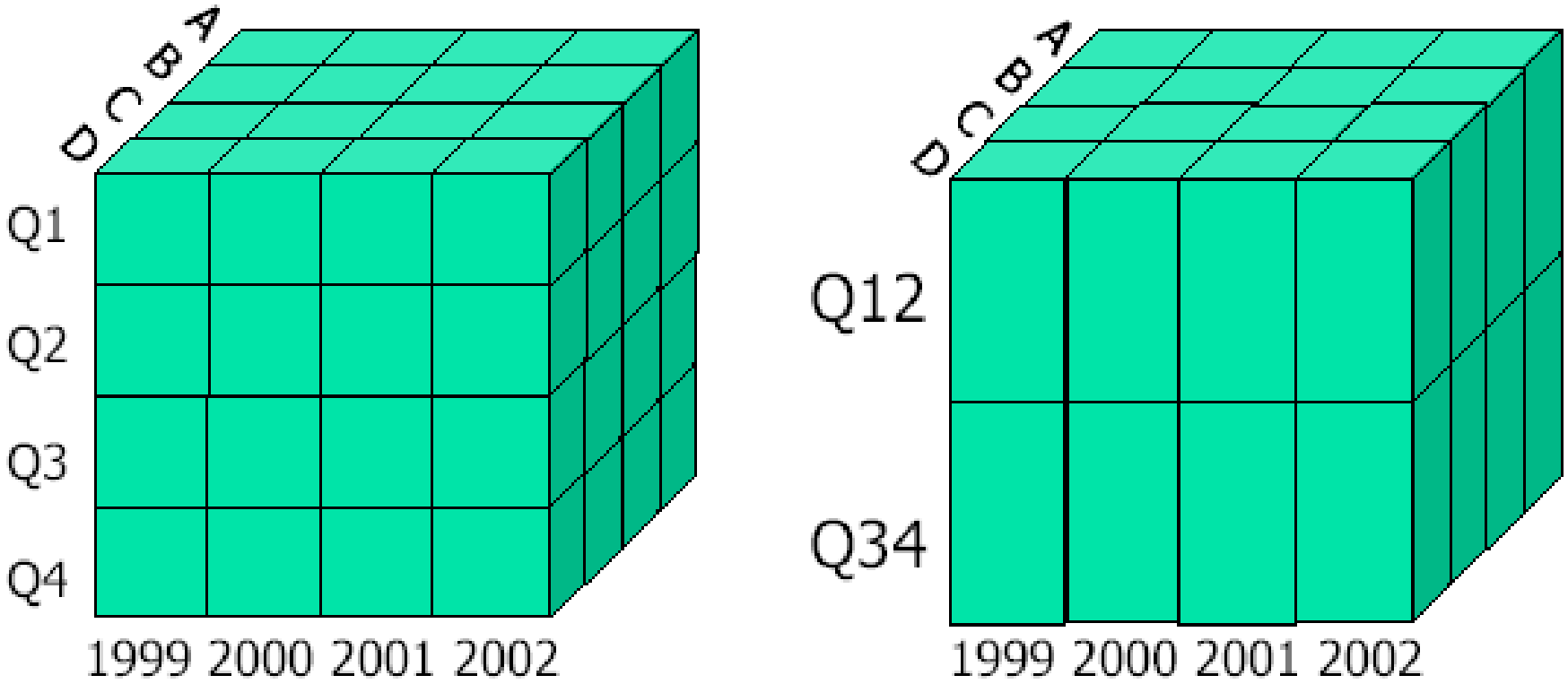
# Strategi Reduksi Data

- Suatu data warehouse bisa menyimpan terabytes data
  - Analisis/menambang data kompleks bisa membutuhkan waktu sangat lama untuk dijalankan pada data set komplit (tak efisien)
- Reduksi data
  - Mengurangi ukuran data set tetapi menghasilkan hasil analitis yang sama (hampir sama)
- Strategi reduksi data
  - Agregasi kubus data
  - Reduksi dimensionalitas—menghilangkan atribut tak penting
  - Kompresi data
  - Reduksi Numerosity reduction—mencocokkan data kedalam model
  - Diskritisasi dan pembuatan konsep hierarki

# Agregasi Kubus Data

- Level terendah dari suatu kubus data
  - Data agregasi data untuk suatu individu entitas yang diminati
  - Misal, suatu customer dalam suatu DW phone calling.
- Banyak level agregasi dalam kubus data
  - Pengurangan ukuran data yang diurusi berikutnya
- Rujukan level yang sesuai
  - Menggunakan representasi terkecil yang cukup untuk memecahkan tugasnya
- Query yang berkaitan dengan agregasi informasi harus dijawab menggunakan kubus data, bila mungkin

# Agregasi Kubus Data

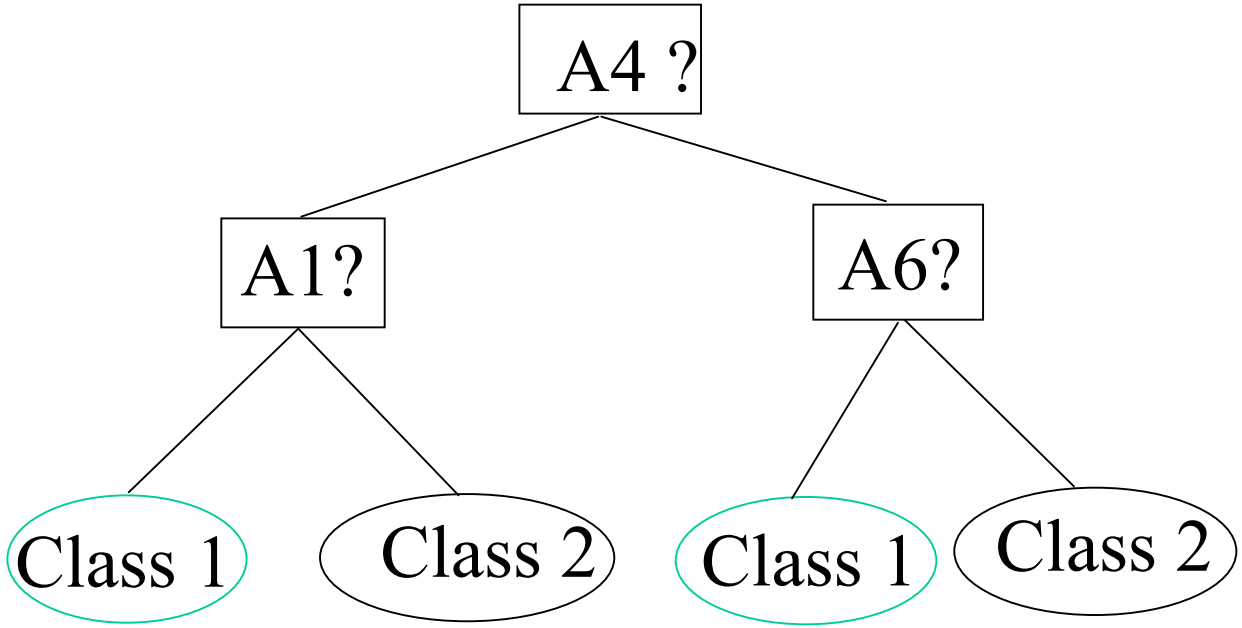


# Reduksi Dimensionalitas

- Fitur seleksi(i.e., attribute subset selection):
  - Memilih sekumpulan fitur minimum sedemikian hingga distribusi peluang dari kelas berbeda bila nilai-nilai fitur tersebut diberikan adalah sedekat mungkin dengan distribusi asli bila nilai-nilai diberikan pada seluruh fitur
  - Mengurangi jumlah pola dalam pola, lebih mudah dipahami
- Metoda heuristik(due to exponential # of choices):
  - Seleksi step-wise forward
  - Eliminasi step-wise backward
  - Kombinasi seleksi forward dan eliminasi backward
  - Induksi pohon keputusan

# Contoh Induksi Pohon Keputusan

Himpunan atribut awal:
{A1, A2, A3, A4, A5, A6}




------> Reduksi himpunan atribut: {A1, A4, A6}

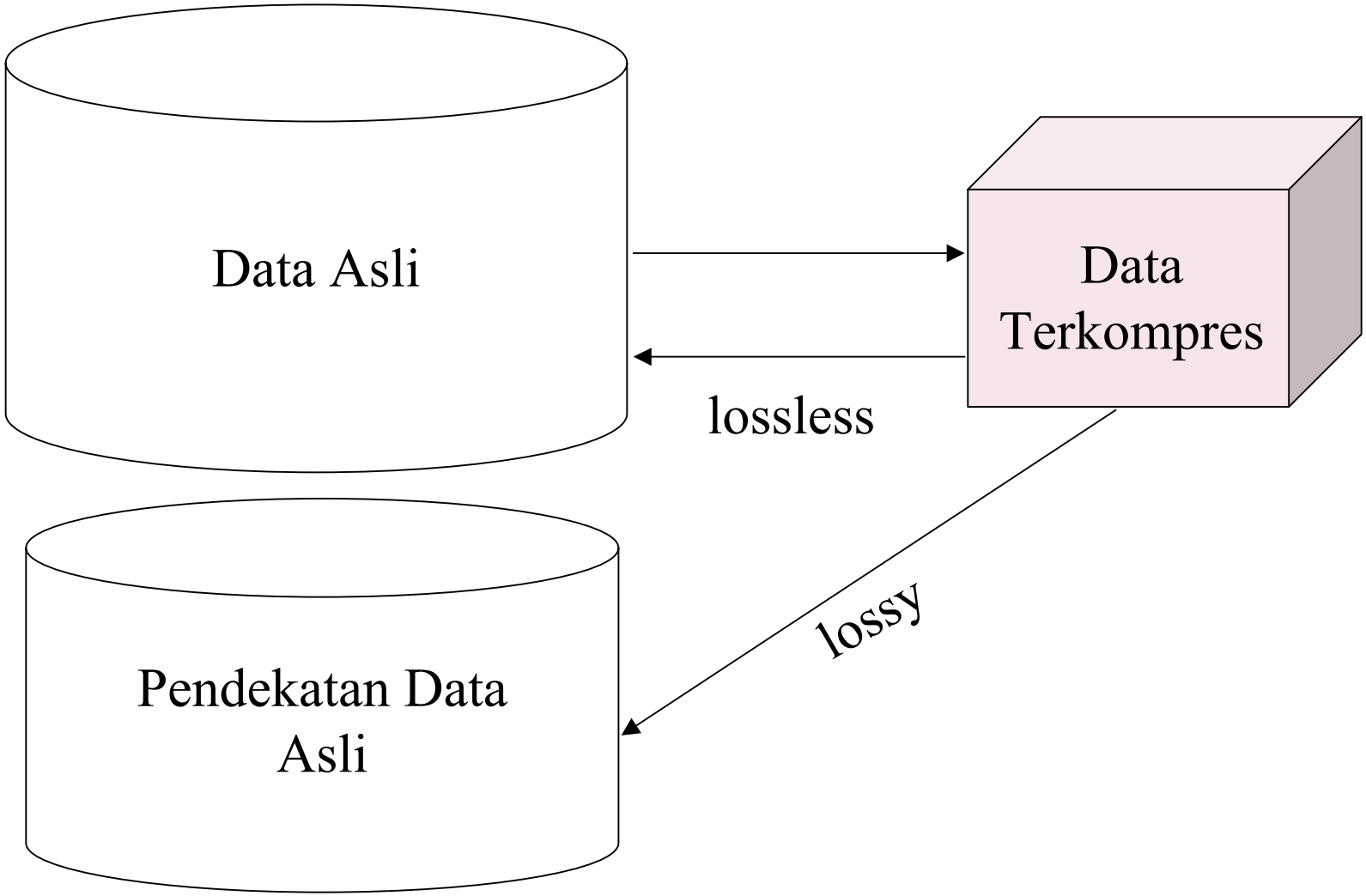
# Fitur Seleksi Metoda Heuristik

- Ada sebanyak $2^d$ sub-fitur yang mungkin dari $d$ fitur
- Beberapa fitur seleksi metoda heuristik:
  - Fitur tunggal terbaik dibawah asumsi fitur bebas: pilih dengan uji berarti.
  - Fitur seleksi step-wise terbaik:
    - Fitur tunggal terbaik dipilih pertama kali
    - Lalu kondisi fitur terbaik untuk yang pertama…
  - Fitur eliminasi step-wise:
    - Secara berulang-ulang menghilangkan fitur yang buruk
  - Kombinasi fitus seleksi dan eliminasi terbaik:
  - Branch and bound optimal:
    - Menggunakan fitur eliminasi dan backtracking

# Kompresi Data

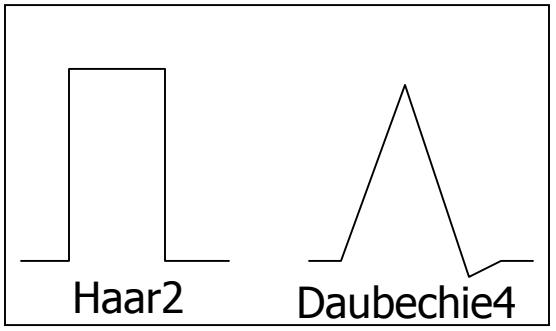
- Kompresi string
  - Ada banyak teori dan algoritma yang telah diselaraskan dengan baik
  - Biasanya lossless
  - Tetapi hanya manipulasi terbatas yang mungkin tanpa perluasan
- Kompresi audio/video
  - Biasanya kompresi lossy, dengan penghalusan progresif
  - Kadang-kadang fragmen kecil dari sinyal bisa direkonstruksi tanpa rekronstruksi keseluruhan
- Urutan waktu bukanlah video
  - Biasanya pendek dan bervariasi lambat menurut waktu

# Kompresi Data



# Transformasi Wavelet

- Discrete wavelet transform (DWT): pemrosesan sinyal linier, analisis multiresolusional




- Pendekatan terkompres: menyimpan hanya suatu bagian kecil dari yang terkuat dari koefisien wavelet
- Mirip dengan Discrete Fourier transform (DFT), tetapi kompresi lossy yang lebih baik, dilokalisasi dalam ruang

# Transformasi Wavelet

- Metoda:
  - Panjang, L, haruslah suatu integer pangkat 2 (diisi dengan 0, bila diperlukan)
  - Setiap transformasi memiliki 2 fungsi: penghalusan dan beda
  - Diterapkan pada pasangan data, yang menghasilkan 2 set data dari panjang L/2
  - Memakai 2 fungsi secara rekursif, sampai panjang yang diinginkan tercapai

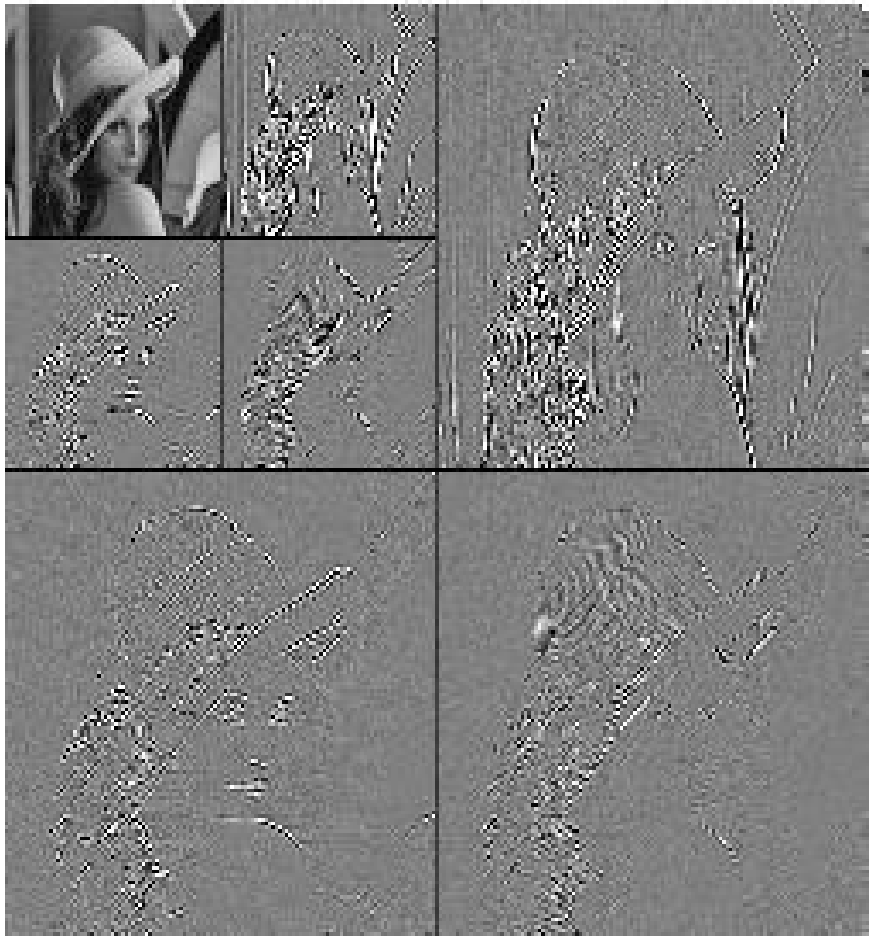
# DWT Untuk Kompresi Citra

- Citra

Low Pass High Pass

Low Pass High Pass

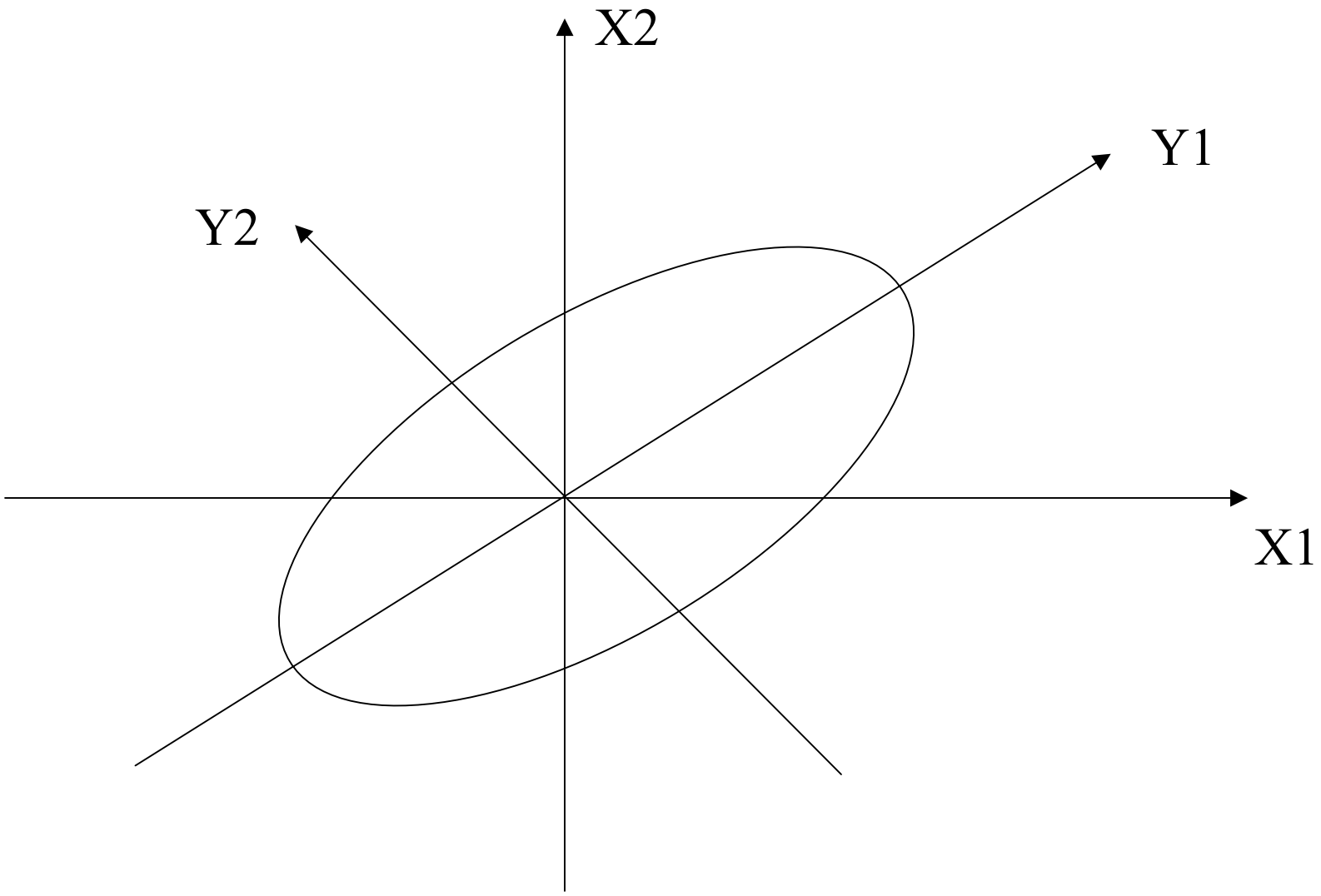
Low Pass High Pass



# Analisis Komponen Utama

- Diberikan N vektor data dari k-dimensi, cari vektor-vektor ortogonal $c <= k$ yang paling baik digunakan untuk menyajikan data
  - Himpunan data asli dikurangi menjadi himpunan yang memuat N vektor-vektor data pada komponen utama c (mengurangi dimensi)
- Setiap vektor data adalah suatu kombinasi linier dari vektor-vektor komponen utama c
- Berlaku hanya untuk data numerik
- Digunakan ketika jumlah dimensi besar

# Analisis Komponen Utama



# Reduksi Numerositi

- Metoda parametrik
  - Misalkan data sesuai dengan suatu model, taksir parameter-parameter model, simpan hanya parameter-parameter tersebut, dan buang datanya (kecuali outlier yang mungkin)
  - Model-model log-linear: dapatkan nilai-nilai pada suatu titik dalam ruang m-D sebagai perkalian atas ruang bagian marginal yang sesuai
- Metoda non-parametrik
  - Tidak memandang model
  - Keluarga utama: histogram, clustering, sampling

# Model Regresi dan Log-linier

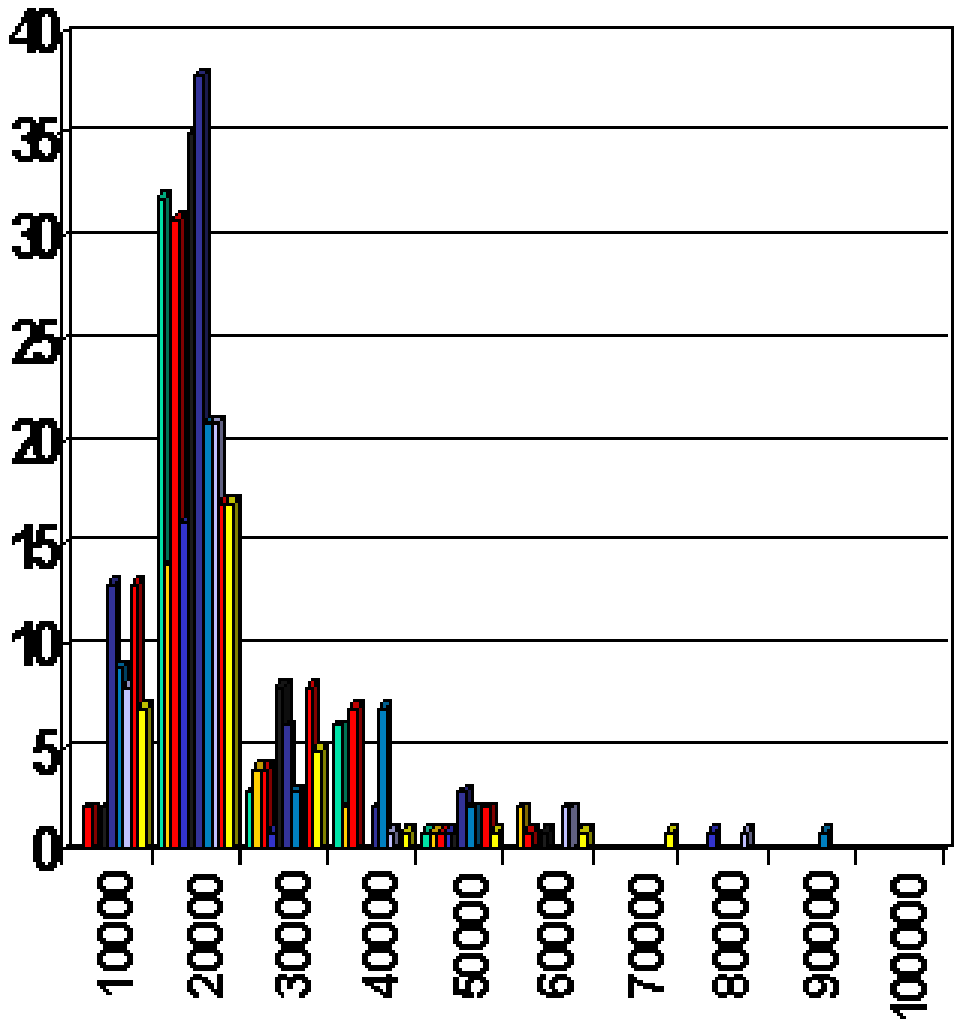
- Regresi linier: Data dimodelkan agar masuk kedalam suatu garis lurus
  - Sering menggunakan metoda least-square agar memenuhi garis tersebut
- Regresi ganda: memungkinkan suatu variabel respon Y untuk dimodelkan sebagai suatu fungsi linier dari vektor fitur multidimensional
- Model log-linear: mendekati distribusi peluang multidimensional diskrit

# Analisa Regresi dan Model log-linier

- Regresi linear: $Y = \alpha + \beta X$
  - 2 parameter , $\alpha$ dan $\beta$ yang menentukan garis tersebut dan akan ditaksir menggunakan data yang dimiliki.
  - Menggunakan kriteria least squares untuk nilai-nilai yang diketahui dari $Y_1, Y_2, \ldots, X_1, X_2, \ldots$.
- Regresi ganda: $Y = b0 + b1\ X1 + b2\ X2$.
  - Berbagai fungsi nonlinier bisa ditransformasikan ke regresi linier.
- Model log-linear:
  - Tabel multi-way dari peluang gabungan didekati dengan suatu perkalian dari tabel-tabel orde lebih rendah.
  - Peluang: $p(a, b, c, d) = \alpha_{ab} \beta_{ac} \chi_{ad} \delta_{bcd}$

# Histogram

- Teknik reduksi data populer
- Membagi data kedalam ember-ember dan menyimpan rata-rata (jumlah) untuk setiap ember
- Bisa dibangun secara optimal dalam satu dimensi menggunakan pemrograman dinamis
- Terkait dengan problem kuantisasi



# Histogram

- Contoh:
  Dataset: 1,1,1,1,1,1,1,1,2,2,2,2,3,3,4,4,5,5,6,6,
  7,7,8,8,9,9,10,10,11,11,12,12
  Histogram: (range, count, sum)
  (1-2,12,16), (3-6,8,36), (7-9,6,48), (10-12,6,66)
- Histogram lebar sama
  - Membagi domain dari suatu atribut kedalam k interval dengan ukuran sama
  - Lebar interval = (Max – Min)/k
  - Secara komputasi mudah
  - Problem dengan skew data dan outliers

# Histogram

- Contoh:
  Dataset: 1,1,1,1,1,1,1,1,2,2,2,2,3,3,4,4,5,5,6,6,
  7,7,8,8,9,9,10,10,11,11,12,12
  Histogram: (range, count, sum)
  (1-3,14,22), (4-6,6,30), (7-9,6,48), (10-12,6,66)
- Histogram kedalaman sama
  - Membagi domain dari suatu atribut kedalam k interval, masing-masing interval memuat jumlah record yang sama
  - Variabel lebar interval
  - Komputasinya mudah

# Histogram

- Contoh:
  Dataset: 1,1,1,1,1,1,1,1,2,2,2,2,3,3,4,4,5,5,6,6,
  7,7,8,8,9,9,10,10,11,11,12,12
  Histogram: (range, count, sum)
  (1,8,8), (2-4,8,22), (5-8,8,52), (9-12,8,84)

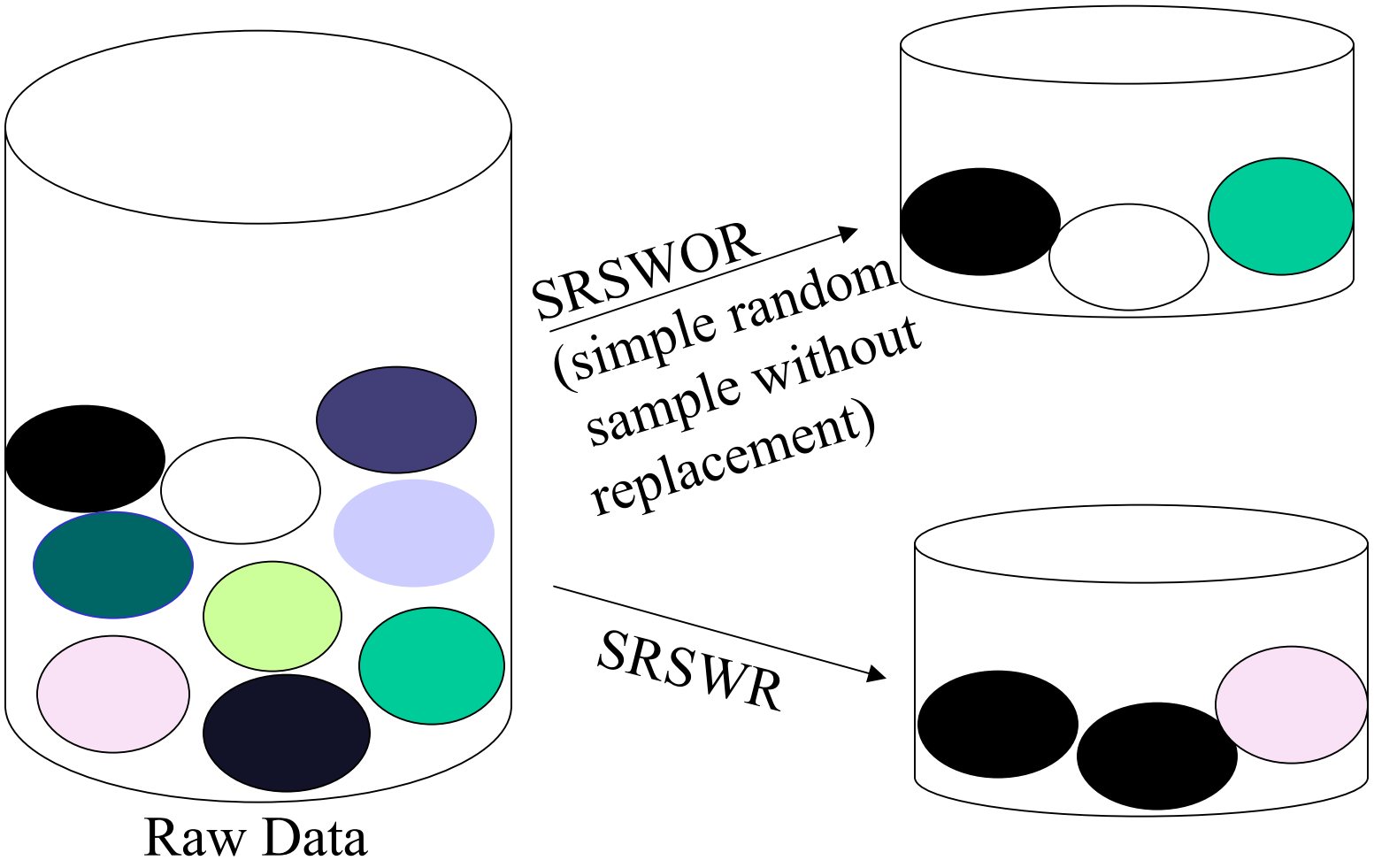
# Clustering

- Mempartisi data set kedalam cluster-cluster, dan bisa hanya menyimpan representasi cluster
- Bisa sangat efektif jika data di-cluster tetapi tidak jika data “dirusak”
- Bisa memiliki clustering hierarki dan bisa disimpan didalam struktur pohon indeks multi-dimensional
- Ada banyak pilihan dari definisi clustering dan algoritma clustering.

# Sampling

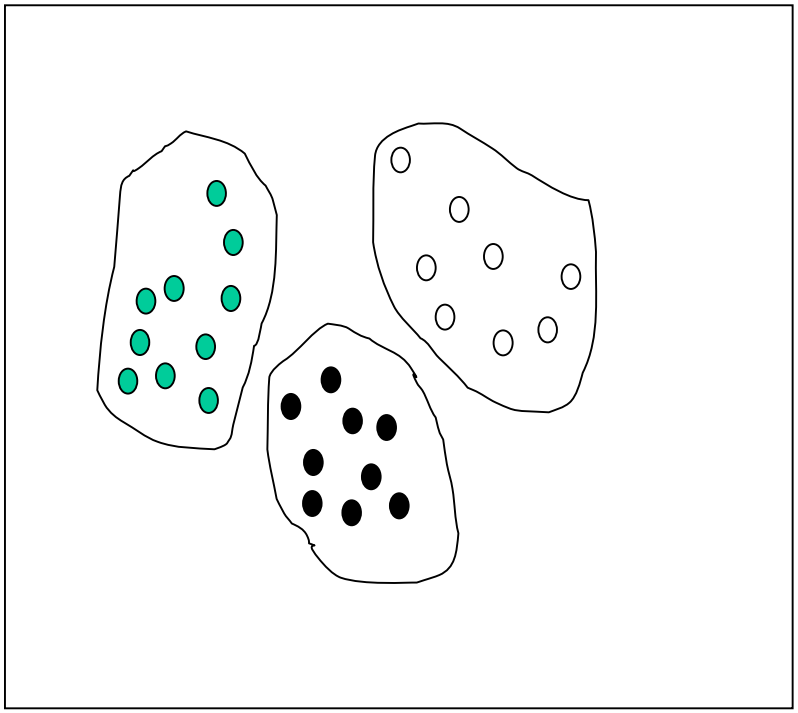
- Memungkinkan suatu algoritma mining untuk dijalankan dalam kompleksitas yang berpotensi sub-linier terhadap ukuran data
- Pilih suatu perwakilan subset dari data tersebut
  - Sampling acak sederhana bisa memiliki kinerja yang sangat buruk bila ada skew
- Kembangkan metoda-metoda sampling adaptif
  - Stratifikasi sampling:
    - Dekati persentasi dari masing-masing kelas (atau subpopulasi yang diminati) dalam database keseluruhan
    - Digunakan dalam hubungannya dengan skewed data
- Sampling bisa tidak mengurangi I/O database (halaman pada suatu waktu).

# Sampling

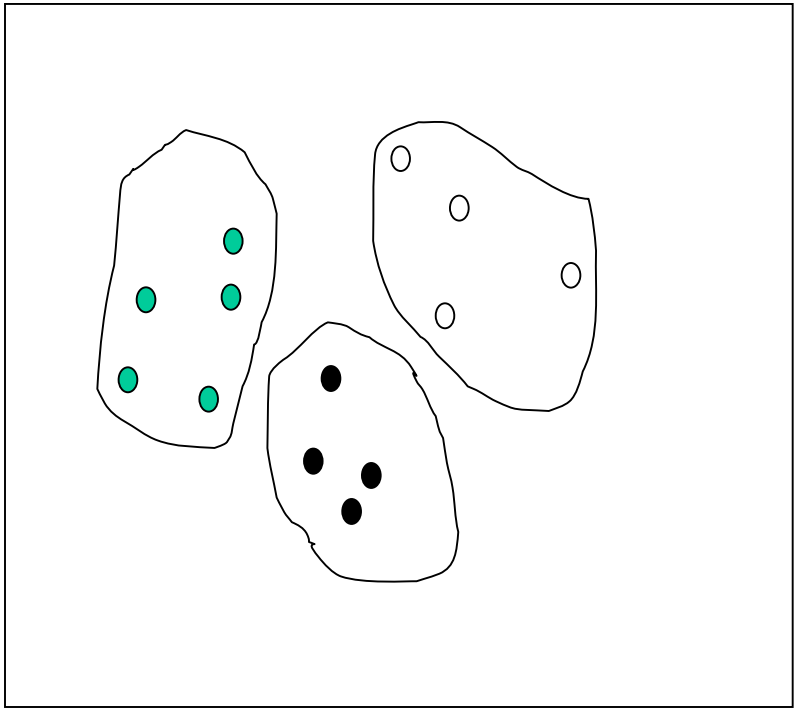

SRSWOR
(simple random
sample without
replacement)
SRSWR
Raw Data

# Sampling

Raw Data



Cluster/Stratified Sample



# Reduksi Hierarki

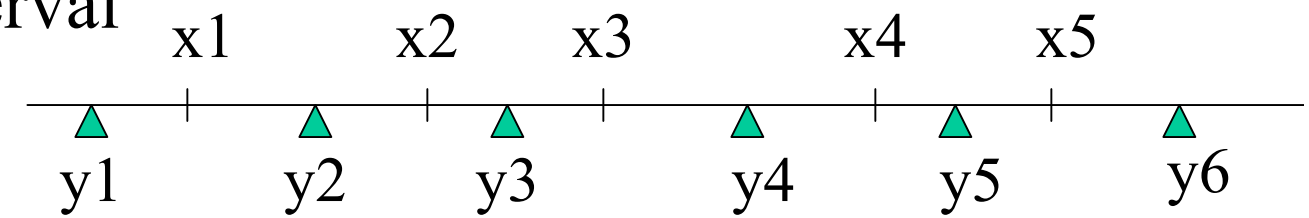
- Gunakan struktur multi resolusi dengan derejat reduksi berbeda
- Clustering hierarkikal sering dilakukan tetapi cenderung mendefinisikan partisi data sets ketimbang "cluster"
- Metoda paramaterik biasanya tidak dapat diuji untuk representasi hierarkikal
- Agregasi hierarkikal
  - Suatu pohon indeks hierarkikal membagi suatu data set kedalam partisi-partisi dengan memberi nilai range dari beberapa atribut
  - Setiap partisi bisa dipandang sebagai suatu ember
  - Jadi suatu pohon indeks dengan agregasi yang disimpan pada setiap node adalah suatu histogram hierarkikal

# Diskritisasi

- Konsep sama dengan histogram
- Membagi domain dari suatu atribut numerik kedalam interval-interval.
- Menggantikan nilai atribut dengan label untuk interval.
- Contoh:
  - Dataset (age; salary):
    (25;30,000),(30;80,000),(27;50,000),
    (60;70,000),(50;55,000),(28;25,000)
  - Dataset diskrit(age, discretizedSalary):
    (25,low),(30,high),(27,medium),(60,high),
    (50,medium),(28,low)

# Diskritisasi

- 3 tipe dari atribut:
  - Nominal — nilai-nilai dari sekumpulan tak berurut
  - Ordinal — nilai-nilai dari suatu himpunan terurut
  - Continuous — bilangan-bilangan riil
- Diskritisasi:
  - Membagi range dari suatu atribut kontinu kedalam interval-interval




  - Beberapa algoritma klasifikasi hanya menerima atribut kategorikal.
  - Mengurangi ukuran data dengan diskritisasi
  - Menyiapkan data untuk analisis lebih lanjut

# Diskritisasi dan Konsep Hierarki

- Diskritisasi
  - Mengurangi jumlah nilai-nilai dari atribut kontinu yang diberikan dengan membagi range atribut kedalam interval-inteval. Label-label interval kemudian bisa digunakan untuk menggantikan nilai-nilai data sesungguhnya
- Konsep hierarki
  - Mengurangi data melalui pengumpulan dan penggantian konsep level rendah (seperti nilai-nilai numerik untuk numerik usia) dengan konsep level lebih tinggi (seperti muda, middle-aged, atau senior)

# Diskritisasi dan Konsep Hierarki Pembuatan Data Numerik

- Binning
- Analisis Histogram
- Analisis Clustering
- Diskritisasi berbasis entropy
- Segmentasi dengan partisi alami

# Diskritisasi Berbasis Entropi

- Diberikan suatu himpunan sampel S, jika S dipartisi kedalam 2 interval S1 dan S2 menggunakan batas T, entropi setelah partisi adalah

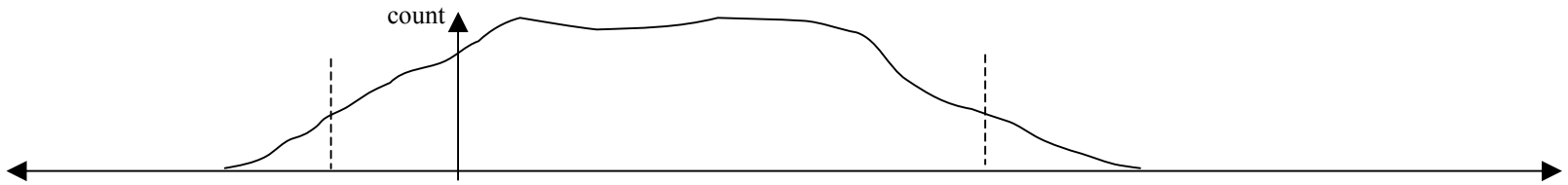
$$E(S,T) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

- Batas yang meminimisasi fungsi entropi atas seluruh batas-batas yang mungkin dipilih sebagai suatu diskritisasi biner.
- Proses ini secara rekursif diterapkan pada partisi yang diperoleh sampai suatu kriteria penghentian ditemukan, misal, $Ent(S) - E(T,S) > \delta$
- Percobaan-percobaan menunjukkan bahwa bahwa diskritisasi ini bisa mengurangi ukuran data dan memperbaiki akurasi klasifikasi

# Segmentasi Dengan Partisi Alami

- Suatu kaidah 3-4-5 sederhana bisa digunakan untuk memecah data numerik kedalam interval "alami" yang relatif seragam.
  - Jika suatu interval memuat 3, 6, 7 atau 9 nilai-nilai berbeda pada digit paling berarti, partisi range tersebut menjadi 3 interval dengan lebar sama
  - Jika suatu interval memuat 2, 4, atau 8 nilai-nilai berbeda pada digit paling berarti, partisi range tersebut kedalam 4 interval
  - Jika suatu interval memuat 1, 5, atau 10 nilai-nilai berbeda pada digit paling berarti, partisi range kedalam 5 interval

# Contoh Kaidah 3-4-5

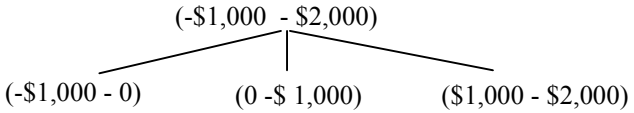



Step 1: -$351 -$159 profit $1,838 $4,700

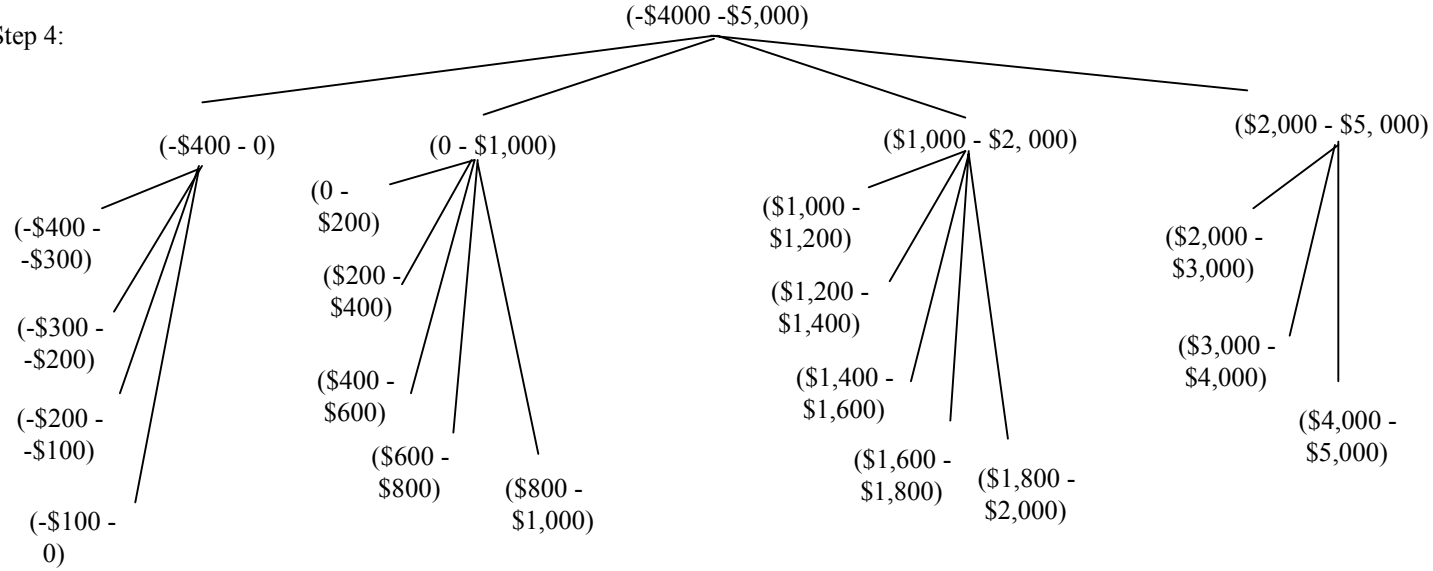
Min Low (i.e, 5%-tile) High(i.e, 95%-0 tile) Max

Step 2: msd=1,000 Low=-$1,000 High=$2,000

Step 3:




Step 4:



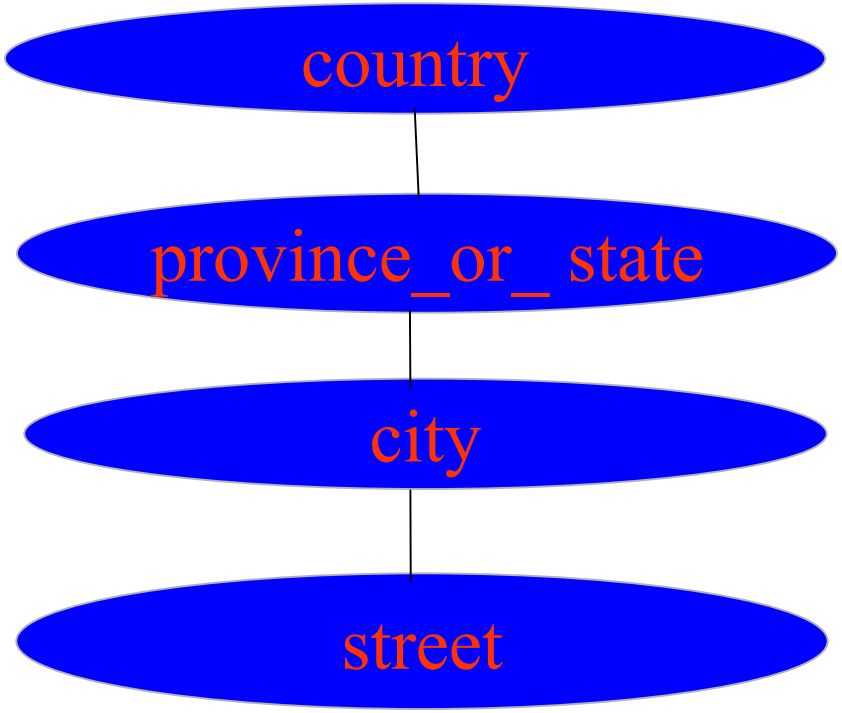
# Pembuatan Konsep Hierarki Untuk Data Kategorikal

- Spesifikasi dari suatu urutan parsial dari atribut secara eksplisit pada level skema oleh user atau pakar
  - street<city<state<country
- Spesifikasi dari suatu bagian dari suatu hierarki dengan pengelompokan data secara eksplisit
  - {Urbana, Champaign, Chicago}<Illinois
- Spesifikasi dari suatu himpunan atribut.
  - Sistem secara otomatis membangun urutan parsial dengan menganalisa jumlah nilai-nilai berbeda
  - Misal, street < city <state < country
- Spesifikasi dari hanya suatu himpunan parsial dari atribut
  - misal, hanya street < city, bukan lainnya

# Pembuatan Konsep Hierarki Otomatis

- Beberapa konsep hierarki bisa secara otomatis dibangun berdasarkan pada analisis dari jumlah nilai-nilai berbeda per atribut dalam data set yang diberikan
  - Atribut dengan nilai-nilai paling berbeda diletakkan pada level terbawah dari hierarki
  - Catatan: Pengecualian—weekday, month, quarter, year

# Pembuatan Konsep Hierarki Otomatis




15 nilai-nilai berbeda

65 nilai-nilai berbeda

3567 nilai-nilai berbeda

674,339 nilai-nilai berbeda