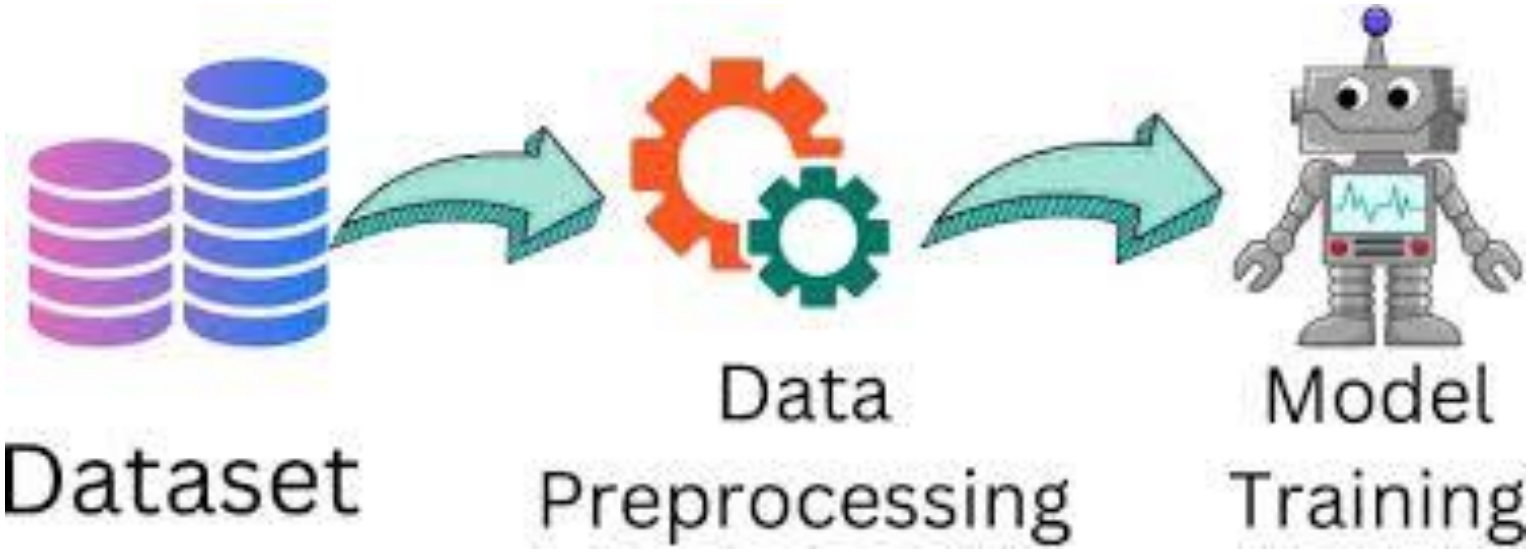
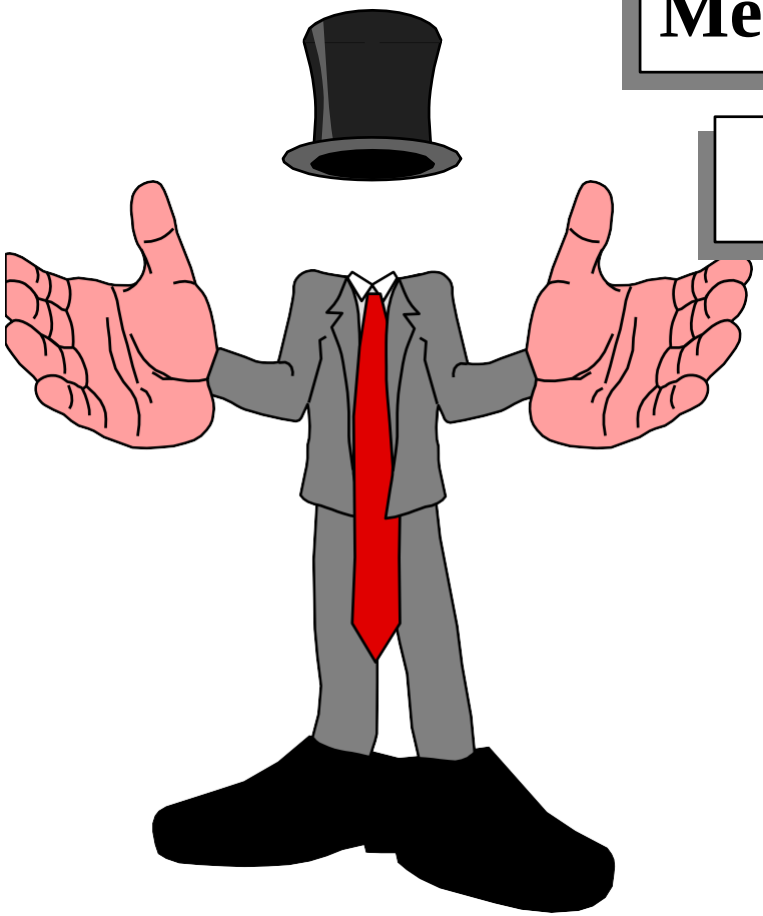
DATA PREPROCESSING

Data Preprocessing

- **Data preprocessing** adalah proses persiapan dan transformasi data mentah menjadi format yang lebih sesuai untuk dianalisis.
- Proses ini bertujuan untuk memastikan kualitas data, menghilangkan kecacatan, dan mengatasi masalah yang mungkin timbul



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?

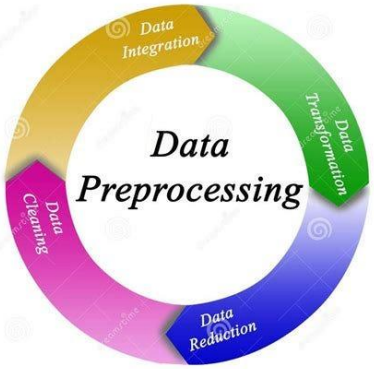
- Kualitas data tidak ada, kualitas hasil mining tidak ada!
  - Kualitas keputusan harus didasarkan kepada kualitas data
    - Misal, duplikasi data atau data hilang bisa menyebabkan ketidak-benaran atau bahkan statistik yang menyesatkan.
  - Data warehouse memerlukan kualitas integrasi data yang konsisten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

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

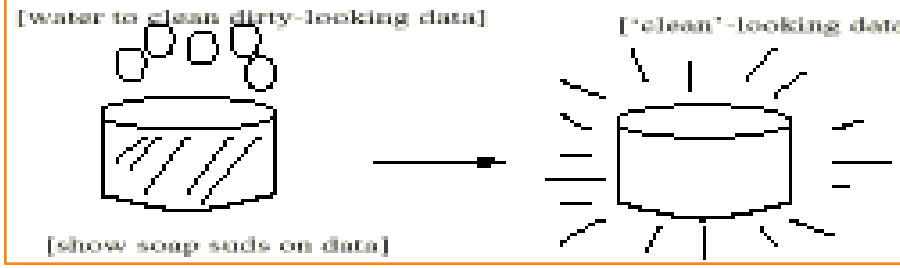
Tugas Utama Pemrosesan Awal Data

- 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

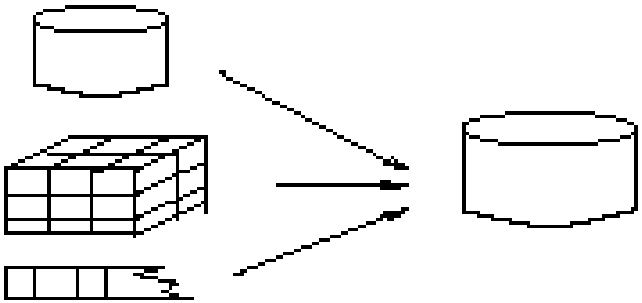


Bentuk-Bentuk Dari Pemrosesan Awal Data

Pembersihan Data



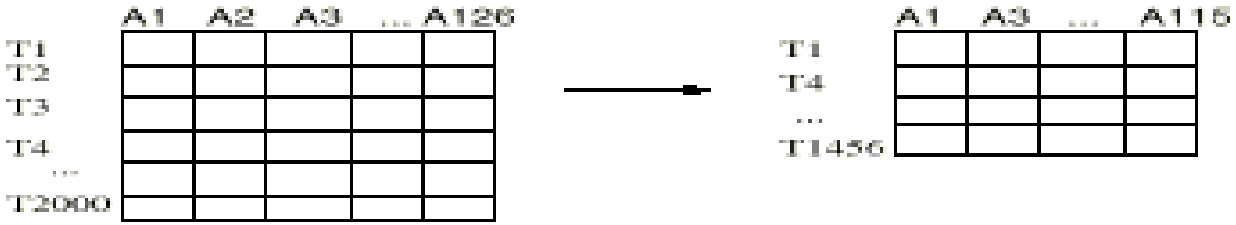
Integrasi Data



Transformasi Data

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

Reduksi Data



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

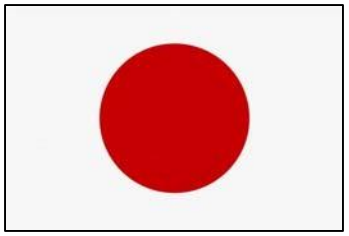
- Problem: integrasi skema heterogen
- Nama-nama tribut berbeda

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