



Direktorat Jenderal Pendidikan Tinggi, Riset, dan, Teknologi
Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi
Republik Indonesia



MICROCREDENTIAL: ASSOCIATE DATA SCIENTIST

01 November – 10 Desember 2021

Pertemuan ke-8

Data Preparation 2: Membersihkan Data & Memvalidasi Data



[ditjen.dikti](https://www.facebook.com/ditjen.dikti)



@ditjendik
ti



[ditjen.dikti](https://www.instagram.com/ditjen.dikti/)



Ditjen
Diktiristek



<https://dikti.kemdikbud.go.id/>

Profil Pengajar: Erwin Eko Wahyudi, S.Kom., M.Cs.



Jabatan Akademik: Tenaga Pengajar

Latar Belakang Pendidikan:

- S1: Ilmu Komputer UGM, 2012-2017
- S2: Ilmu Komputer UGM, 2017-2019

Riwayat/Pengalaman Pekerjaan:

- Dosen, UGM, 2021-sekarang
- AI Engineer Recommender System, Bukalapak, 2019-2021

Contact Pengajar:

Ponsel:

0812 1195 4011

Email:

erwin.eko.w@ugm.ac.id

Referensi: SKKNI Data Science

KODE UNIT : **J.62DMI00.008.1**

JUDUL UNIT : **Membersihkan Data**

DESKRIPSI UNIT: Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam membersihkan data yang sesuai permintaan atau kebutuhan.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Melakukan pembersihan data yang kotor	1.1 Strategi pembersihan data ditentukan berdasarkan hasil telaah data. 1.2 Data yang kotor dikoreksi berdasarkan strategi pembersihan data.
2. Membuat laporan dan rekomendasi hasil membersihkan data	2.1 Masalah dan teknis koreksi data dideskripsikan sesuai dengan kondisi data dan strategi pembersihan data. 2.2 Evaluasi dihasilkan berdasarkan analisis koreksi yang telah dilakukan. 2.3 Evaluasi proses dan hasilnya didokumentasikan.

BATASAN VARIABEL

1. Konteks variabel
 - 1.1 Strategi pembersihan data dapat berupa pengisian dengan nilai yang tepat (*mean, median, min/max, mode, etc*), koreksi nilai standar, diisi dengan konstanta, menghapus baris kosong dan lain-lain.
 - 1.2 Data yang kotor dapat berupa data terstruktur maupun tidak terstruktur berupa *missing value*, data yang salah, dan *data outlier*.
 - 1.3 Rekomendasi adalah tindak lanjut dari proses pembersihan data.
 - 1.4 Permintaan atas kebutuhan disesuaikan dengan *standard* di organisasi terkait.



Referensi: SKKNI Data Science

KODE UNIT : **J.62DMI00.006.1**

JUDUL UNIT : **Memvalidasi Data**

DESKRIPSI UNIT: Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam memvalidasi data untuk *data science*.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Melakukan pengecekan kelengkapan data	1.1 Penilaian kualitas data dari hasil telaah disajikan sesuai tujuan teknis <i>data science</i> . 1.2 Penilaian tingkat kecukupan data dari hasil telaah disajikan sesuai tujuan teknis <i>data science</i> .
2. Membuat rekomendasi kelengkapan data	2.1 Rekomendasi hasil penilaian kualitas disusun sesuai tujuan teknis <i>data science</i> 2.2 Rekomendasi hasil penilaian kecukupan data disusun sesuai tujuan teknis <i>data science</i> .

BATASAN VARIABEL

1. Konteks variabel
 - 1.1 Kualitas data merupakan data yang tidak memiliki nilai mencurigakan, kosong, inkonsisten, duplikasi maupun ambigu.
 - 1.2 Tingkat kecukupan data merupakan batasan dari tujuan teknis *data science* terkait kualitas data.
 - 1.3 Rekomendasi hasil penilaian kualitas data adalah usulan terkait kebenaran, kelengkapan dan konsistensi data.
 - 1.4 Rekomendasi hasil penilaian kecukupan data adalah usulan untuk cukup atau perlu diulang pengumpulan datanya.



Course Definition

- Bagian Kedua dari Tiga materi Data Preparation
- Berfokus pada Pembersihan Data dan validasi data Konteksnya yaitu:
 - strategi pembersihan data kotor (noise, bias, missing value, outlier, dll)
 - pengecekan kualitas dan tingkat kecukupan data
- Pengetahuan dan pemahaman akan data preparation menjadi syarat mutlak untuk menghasilkan model prediksi yang optimal.



Learning Objective

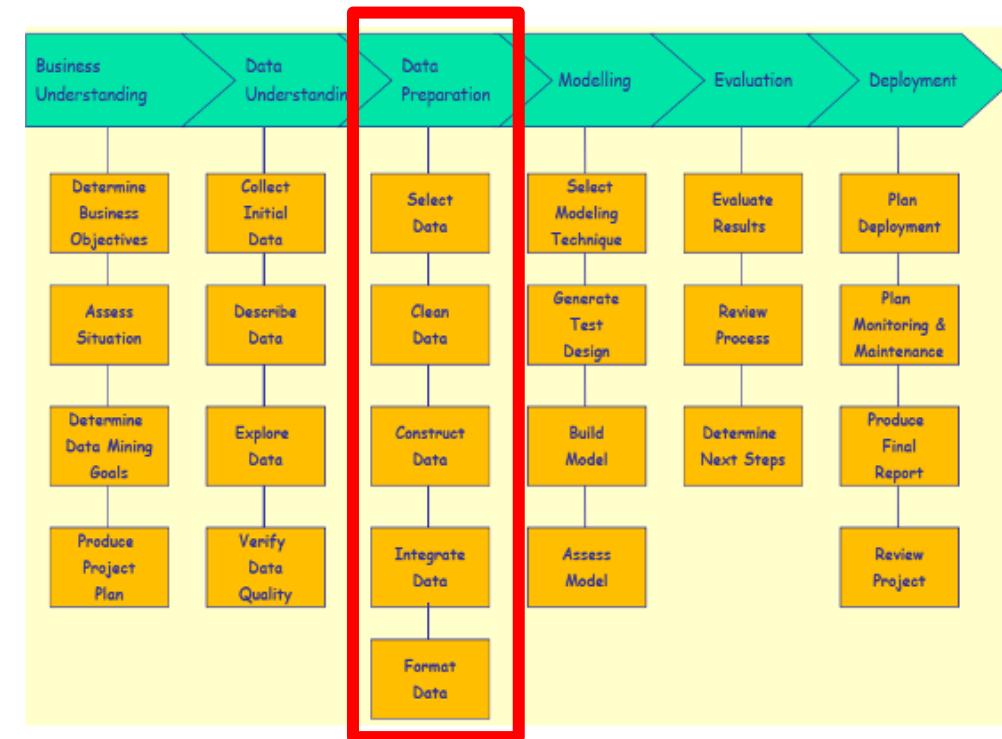
Dalam pelatihan ini diharapkan:

- Peserta mampu melakukan pembersihan data
- Peserta mampu melakukan memeriksa (cek) kualitas dan kecukupan data



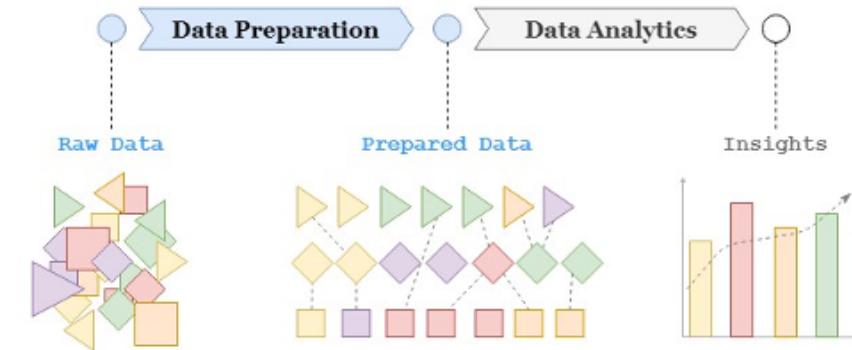
Data Preparation dalam CRISP-DM

- Akronim dari: **CRoss Industry Standard Process Data Mining**
- Metodologi umum untuk data mining, analitik, dan proyek data sains, berfungsi menstandarkan proses data mining lintas industri
- Digunakan untuk semua level dari pemula hingga pakar



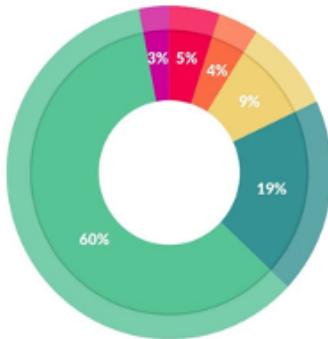
Terminologi dan Definisi

- Istilah lain: **Data Pre-processing**, **Data Manipulation**, **Data Cleansing/Normalization**
- Definisi:
 - *transformasi data mentah menjadi format yang mudah dipahami*
 - menemukan data yang relevan untuk disertakan dalam aplikasi analitik shg memberikan informasi yang dicari oleh analis atau pengguna bisnis
 - *langkah pra-pemrosesan yang melibatkan pembersihan, transformasi, dan konsolidasi data.*



- Definisi:
 - proses yang melibatkan koneksi ke satu atau banyak sumber data yang berbeda, membersihkan data kotor, memformat ulang atau merestrukturasi data, dan akhirnya menggabungkan data ini untuk digunakan untuk analisis.

Fakta Terkait Data Preparation



What data scientists spend the most time doing

- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

60-80% porsi kegiatan data saintis
(forbes, crowdflower 2016)

- data yang ada saat ini dari banyak sumber data dan format yang beragam (terstruktur, semi, dan tidak terstruktur)
- kualitas model prediktif bergantung pada kualitas data (GIGO)

Data Preparation Matters

65% of organizations said it is **very important to simplify making information available**. The most often required big data preparation activities are:



In the analytic process, the tasks in which organizations spend the most time are reviewing data for quality and consistency (52%) and preparing data for analysis (46%).

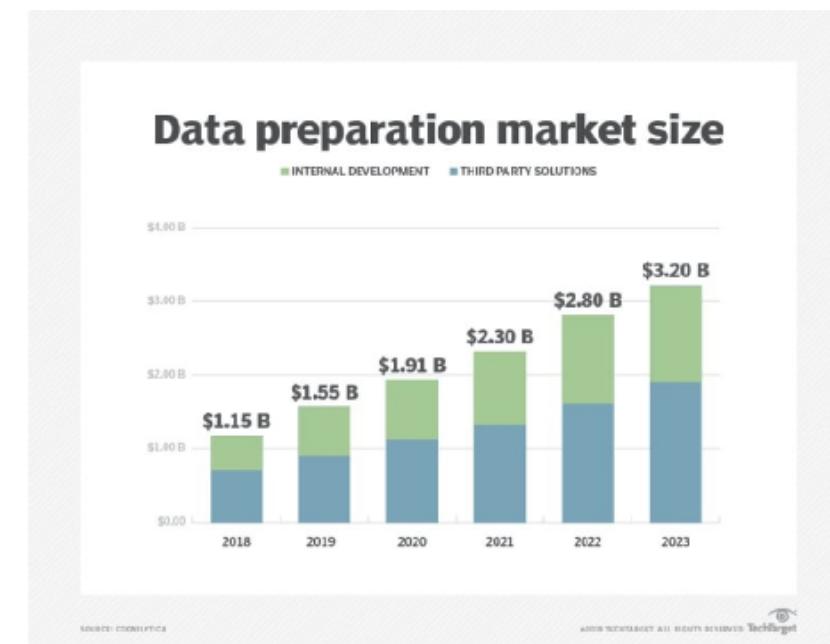
Pentingnya Data Preparation

- data perlu diformat sesuai dengan software yang digunakan
- data perlu disesuaikan dengan metode data sain yang digunakan data real-world cenderung ‘kotor’:
 - tidak komplit: kurangnya nilai attribute, kurangnya atribut tertentu/penting, hanya berisi data aggregate. misal: pekerjaan="" (tidak ada isian)
 - *noisy*: memiliki error atau outlier. misal: Gaji="-10", Usia="222"
- data real-world cenderung ‘kotor’: tidak konsisten: memeliki perbedaan dalam kode dan nama. misal : Usia="32" TglLahir="03/07/2000"; rating "1,2,3" -- > rating "A, B, C"
- kolom dan baris yang saling bertukar banyak variabel dalam satu kolom yang sama



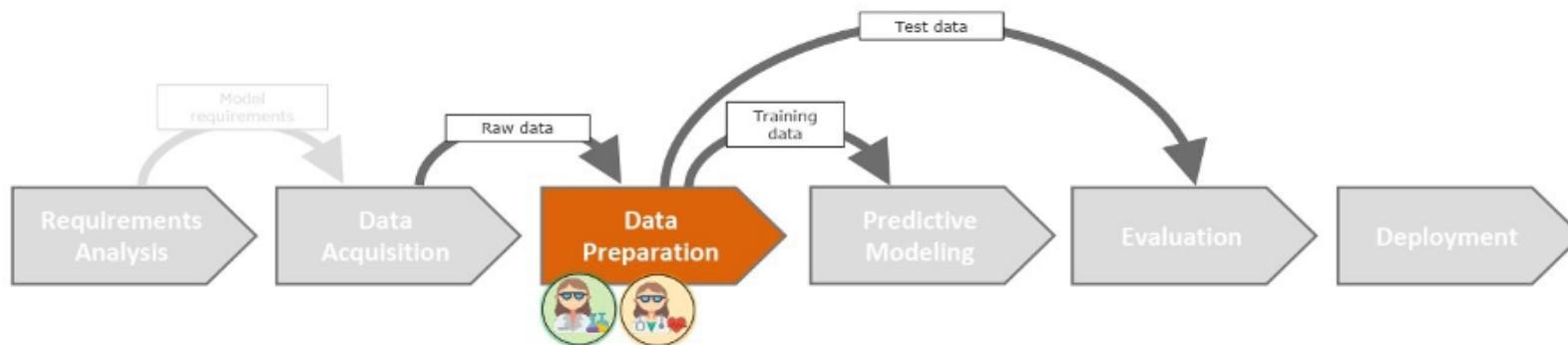
Manfaat Data Preparation

- Kompilasi Data menjadi Efisien dan Efektif (menghindari duplikasi)
- Identifikasi dan Memperbaiki Error
- Mudah Perubahan Secara Global
- Menghasilkan Informasi yang Akurat untuk Pengambilan Keputusan
- Nilai Bisnis dan ROI (Return on Investment) akan Meningkat





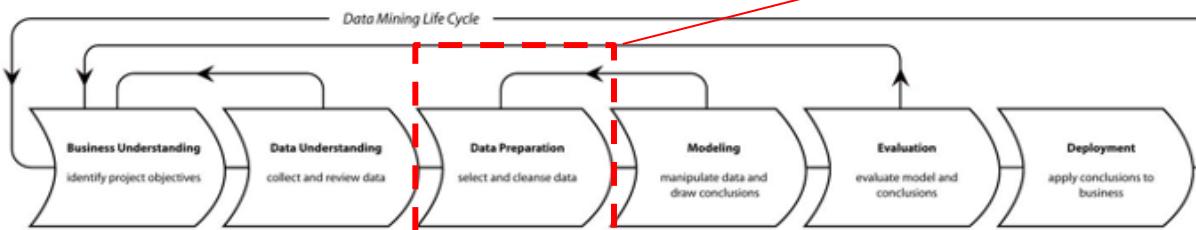
Tahapan dan Tantangan Data Preparation



- Memakan Waktu Lama
- Porsi Teknis yang Dominan
- Data yang Tersedia Tidak Akurat atau Jelas/Tidak Langsung Pakai
- Data tidak Balance Saat Pengambilan Sampel
- Rentan akan Error

Data Preparation dalam CRISP-DM

Phases



Determine Business Objectives
Background
Business Objectives
Business Success Criteria
(Log and Report Process)

Assess Situation
Inventory of Resources,
Requirements, Assumptions,
and Constraints
Risks and Contingencies
Terminology
Costs and Benefits
(Log and Report Process)

Determine Data Mining Goals
Data Mining Goals
Data Mining Success Criteria
(Log and Report Process)

Produce Project Plan
Project Plan
Initial Assessment of Tools and
Techniques
(Log and Report Process)

Generic Tasks
Specialized Tasks
(Process Instances)

Data Mining Life Cycle

Collect Initial Data
Initial Data Collection Report
(Log and Report Process)

Describe Data
Data Description Report
(Log and Report Process)

Explore Data
Data Exploration Report
(Log and Report Process)

Verify Data Quality
Data Quality Report
(Log and Report Process)

Data Set
Data Set Description
(Log and Report Process)

Select Data
Rationale for Inclusion/
Exclusion
(Log and Report Process)

Clean Data
Data Cleaning Report
(Log and Report Process)

Construct Data
Derived Attributes
Generated Records
(Log and Report Process)

Integrate Data
Merged Data
(Log and Report Process)

Format Data
Reformatted Data
(Log and Report Process)

Select Modeling Technique
Modeling Technique
Modeling Assumptions
(Log and Report Process)

Generate Test Design
Test Design
(Log and Report Process)

Build Model Parameter Settings
Models
Model Description
(Log and Report Process)

Assess Model
Model Assessment
Revised Parameter
(Log and Report Process)

Evaluate Results
Align Assessment of Data Mining Results with Business Success Criteria
(Log and Report Process)

Approved Models
Review Process
Review of Process
(Log and Report Process)

Determine Next Steps
List of Possible Actions
Decision
(Log and Report Process)

Plan Deployment
Deployment Plan
(Log and Report Process)

Plan Monitoring and Maintenance
Monitoring and Maintenance Plan
(Log and Report Process)

Produce Final Report
Final Report
Final Presentation
(Log and Report Process)

Review Project
Experience
Documentation
(Log and Report Process)

a visual guide to CRISP-DM methodology

SOURCE CRISP-DM 1.0
<http://www.crisp-dm.org/download.htm>
DESIGN Nicole Leaper
<http://www.nicoleleaper.com>



Tahapan Data Preparation: Pemilihan, Pembersihan & Validasi

1. Pilih/ Select Data

- Pertimbangkan pemilihan data
- Tentukan dataset yang akan digunakan
- Kumpulkan data tambahan yang sesuai (internal atau eksternal)
- Pertimbangkan penggunaan teknik pengambilan sampel

Jelaskan mengapa data tertentu dimasukkan atau dikecualikan

2. Bersihkan/ Clean Data

- Perbaiki, hapus atau abaikan noise
- Putuskan bagaimana menangani nilai-nilai khusus dan maknanya
- Tingkat agregasi, nilai yang hilang (missing value), dll
- Bersihkan atau manipulasi outlier

3. Validasi Data

- Periksa/Nilai Kualitas Data
- Periksa/Nilai Tingkat Kecukupan Data

Referensi: SKKNI Data Science

KODE UNIT : **J.62DMI00.008.1**

JUDUL UNIT : **Membersihkan Data**

DESKRIPSI UNIT: Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam membersihkan data yang sesuai permintaan atau kebutuhan.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Melakukan pembersihan data yang kotor	1.1 Strategi pembersihan data ditentukan berdasarkan hasil telaah data. 1.2 Data yang kotor dikoreksi berdasarkan strategi pembersihan data.
2. Membuat laporan dan rekomendasi hasil membersihkan data	2.1 Masalah dan teknis koreksi data dideskripsikan sesuai dengan kondisi data dan strategi pembersihan data. 2.2 Evaluasi dihasilkan berdasarkan analisis koreksi yang telah dilakukan. 2.3 Evaluasi proses dan hasilnya didokumentasikan.

1. Konteks variabel

- 1.1 Strategi pembersihan data dapat berupa pengisian dengan nilai yang tepat (*mean, median, min/max, mode, etc*), koreksi nilai standar, diisi dengan konstanta, menghapus baris kosong dan lain-lain.
- 1.2 Data yang kotor dapat berupa data terstruktur maupun tidak terstruktur berupa *missing value*, data yang salah, dan *data outlier*.
- 1.3 Rekomendasi adalah tindak lanjut dari proses pembersihan data.
- 1.4 Permintaan atas kebutuhan disesuaikan dengan *standard* di organisasi terkait.

Paramater/Daftar Isi Dokumentasi Data Cleaning

Laporan dokumentasi data cleaning, setidaknya memiliki parameter berikut:

- Data Set Description
- Data Set yang digunakan
- Jenis noise yang terjadi pada data (diantaranya: Missing data, Data errors; Coding inconsistencies; Missing/ bad metadata)
- Pendekatan yang dilakukan untuk menghilangkan noise tersebut
- Teknik mana yang digunakan sehingga berhasil untuk menghilangkan noise tersebut
- Apakah ada kasus atau atribut yang tak dapat diselamatkan
- Pastikan data yang dikecualikan karena kondisi noisenya

Referensi: SKKNI Data Science

KODE UNIT : **J.62DMI00.006.1**

JUDUL UNIT : **Memvalidasi Data**

DESKRIPSI UNIT : Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam memvalidasi data untuk *data science*.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Melakukan pengecekan kelengkapan data	1.1 Penilaian kualitas data dari hasil telaah disajikan sesuai tujuan teknis <i>data science</i> . 1.2 Penilaian tingkat kecukupan data dari hasil telaah disajikan sesuai tujuan teknis <i>data science</i> .
2. Membuat rekomendasi kelengkapan data	2.1 Rekomendasi hasil penilaian kualitas disusun sesuai tujuan teknis <i>data science</i> 2.2 Rekomendasi hasil penilaian kecukupan data disusun sesuai tujuan teknis <i>data science</i> .

1. Konteks variabel

- 1.1 Kualitas data merupakan data yang tidak memiliki nilai mencurigakan, kosong, inkonsisten, duplikasi maupun ambigu.
- 1.2 Tingkat kecukupan data merupakan batasan dari tujuan teknis *data science* terkait kualitas data.
- 1.3 Rekomendasi hasil penilaian kualitas data adalah usulan terkait kebenaran, kelengkapan dan konsistensi data.
- 1.4 Rekomendasi hasil penilaian kecukupan data adalah usulan untuk cukup atau perlu diulang pengumpulan datanya.

Resampling



Paramater/Daftar Isi Dokumentasi Data Validation

Laporan dokumentasi data validasi, setidaknya memiliki parameter berikut:

- Validasi data

- Kebenaran, misal di Indonesia isian Gender yang diakui hanya 2 P/W; Agama hanya 6 (Islam, Protestan, Katholik, Hindu, Budha, Konghucu)
- Kelengkapan, misal data propinsi seluruh Indonesia (34 prov), namun hanya sebagian yg ada
- Konsistensi, misal penulisan STM atau SMK;

- Kecukupan data → Perlukan diulang berikan justifikasi (Resampling)



Rincian Tahapan Data Preparation

3. Bangun/ Construct Data

- Atribut turunan.
- Latar belakang pengetahuan.
- Bagaimana atribut yang hilang dapat dibangun atau diperhitungkan

4. Integrasi/ Integrate Data

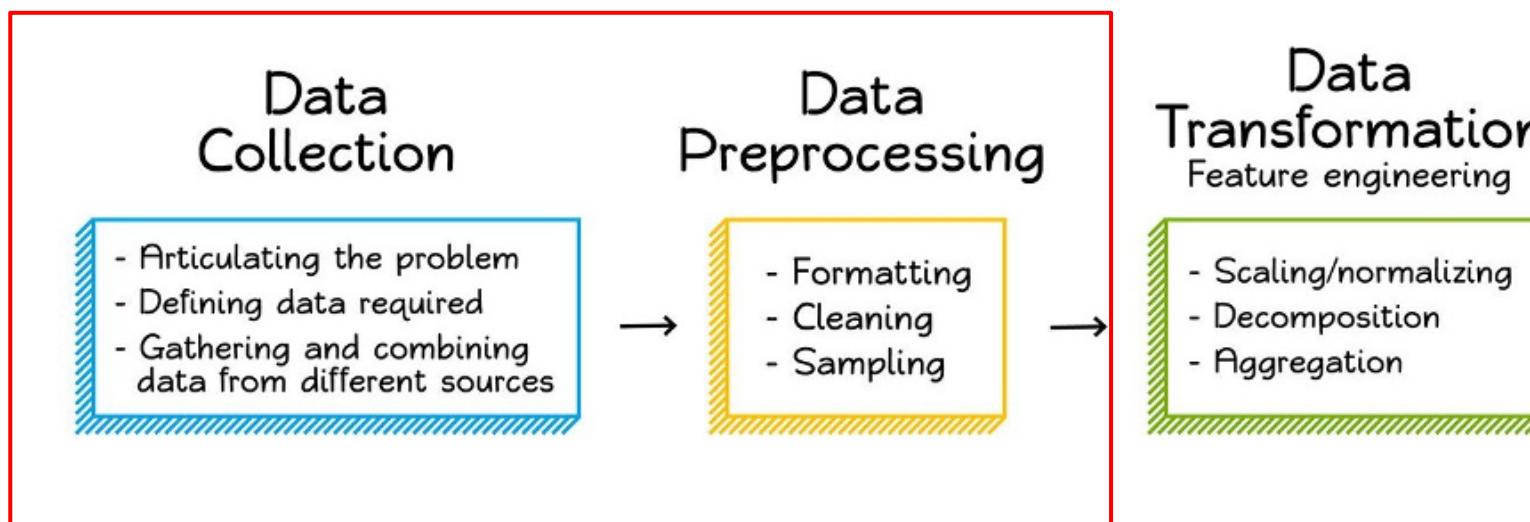
- Mengintegrasikan sumber dan menyimpan hasil (tabel dan catatan baru)

5. Bentuk/ Format Data



Tahapan Data Preparation: Versi Simple

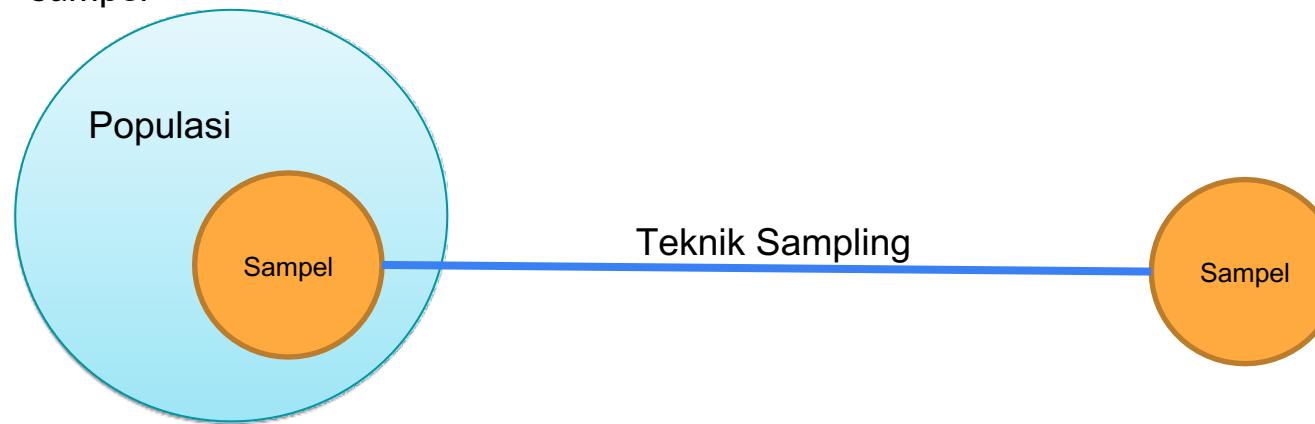
Data Preparation Process





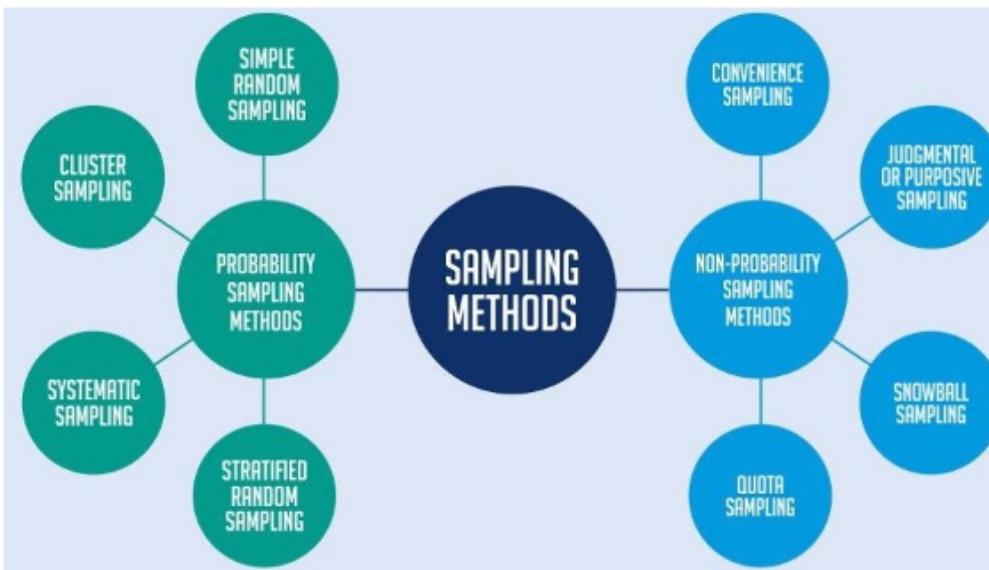
Sampling Data: Pengertian Sampling

- Sebelum melakukan tahapan dalam data preparation, terlebih dahulu adalah pemilihan/penentuan objek yang dapat dilakukan dengan menggunakan penentuan:
 - Populasi
 - Sampel



Sampling Data: Metode Sampling

- Kategori Metode Sampling



- Probability Sampling:
 - Populasi diketahui
 - Randomisasi/keteracakalan: Ya
 - Conclusiver
 - Hasil: Unbiased
 - Kesimpulan: Statistik
- Non-Probability Sampling:
 - Populasi tidak diketahui
 - Keterbatasan penelitian
 - Randomisasi/keteracakalan: Tidak
 - Exploratory
 - Hasil: Biased
 - Kesimpulan: Analitik



Sampling Data: Metode Sampling

When to use probability sampling?

- When you want to reduce the sampling bias**
Probability sampling leads to higher quality findings because it provides an unbiased representation of the population.
- When the population is usually diverse**
This sampling method will help pick samples from various socio-economic strata, background, etc. to represent the broader population.
- To create an accurate sample**
Researchers use proven statistical methods to draw a precise sample size to obtain well-defined data.

Learn more: www.questionpro.com/blog/probability-sampling/

QuestionPro

Types of probability sampling

Simple random sampling

Cluster sampling

Systematic sampling

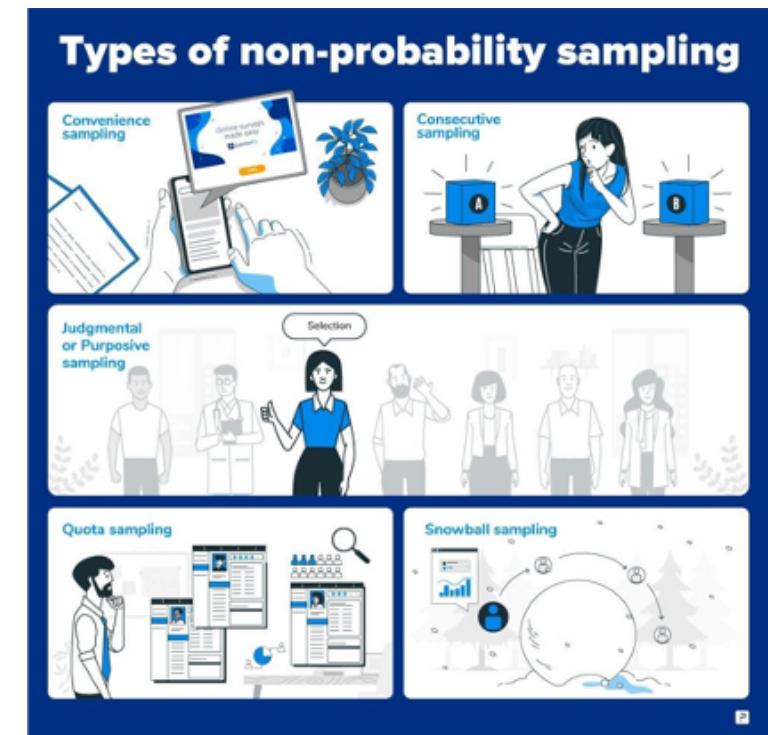
Stratified random sampling

QuestionPro



Sampling Data: Teknik Sampling

Non-Probability Methods





Sampling Data: Tahapan Sampling

Step 1

Identify and define Target Population

Step 2

Select Sampling Frame

Step 3

Choose Sampling methods

Step 4

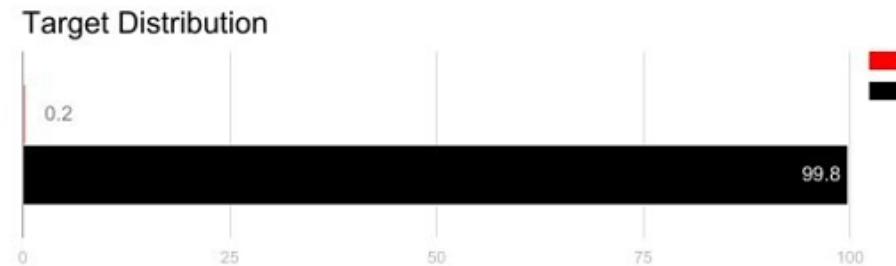
Determine Sample Size

Step 5

Collect the required Data

Imbalance Dataset: Resampling

- Ini dilakukan setelah proses pemilihan, pembersihan dan rekayasa fitur dilakukan atas pertanyaan:
 - Tanya: apakah kelas target data yang kita inginkan telah secara sama terdistribusi di seluruh dataset?
 - Jawab: Di banyak kasus tidak/belum tentu. Biasanya terjadi imbalance (ketidakseimbangan) antara dua kelas. Misal utk dataset tentang deteksi fraud di perbankan, lelang real-time, atau deteksi intrusi di network! Biasanya data dari dataset tersebut berukuran sangat kecil atau kurang dari 1%, namun sangat signifikan. Kebanyakan algoritma ML tidak bekerja baik utk dataset imbalance tsb.



Imbalance Dataset: Resampling

Berikut adalah bbrp cara utk mengatasi imbalance dataset:

Gunakan pengukuran (metrik) yang tepat, misal dengan menggunakan:

- **Precision:** berapa banyak instance yang relevan
- **Recall/Sensitifitas:** berapa banyak instance yang dipilih
- **F1 score:** harmonisasi mean dari precision dan recall
- **Matthews correlation coefficient (MCC):** koefisien korelasi antara klasifikasi biner antara observasi vs prediksi
- **Area under the ROC curve (AUC):** relasi antara tingkat true-positive vs false-positive

Resample data training, dengan dua metode:

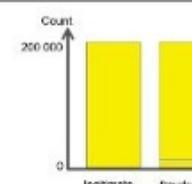
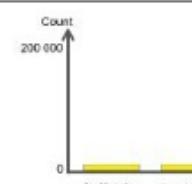
- **Undersampling:** menyeimbangkan dataset dengan mereduksi ukuran kelas yang melimpah.
Dilakukan jika kuantitas data mencukupi
- **Oversampling:** Kebalikan dari undersampling, dilakukan jika kuantitas data tidak mencukupi

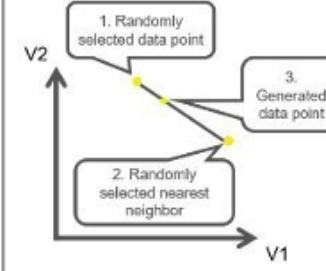
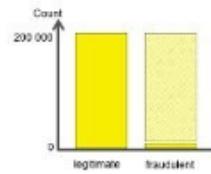
Imbalance Dataset: Resampling

- Teknik

Resampling (SMOTE)

- oversampling (Bootstrap)
- undersampling (Bootstrap)

Oversampling (Bootstrap)	Randomly draw with replacement a sample of fraudulent transactions until the number of fraudulent transactions is ca equal to the number of legitimate transactions	 <table border="1"> <thead> <tr> <th>Transaction Type</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>legitimate</td> <td>~200,000</td> </tr> <tr> <td>fraudulent</td> <td>~200,000</td> </tr> </tbody> </table>	Transaction Type	Count	legitimate	~200,000	fraudulent	~200,000
Transaction Type	Count							
legitimate	~200,000							
fraudulent	~200,000							
Undersampling (Bootstrap)	Randomly draw with replacement as many legitimate transactions as there are fraudulent transactions	 <table border="1"> <thead> <tr> <th>Transaction Type</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>legitimate</td> <td>< 10,000</td> </tr> <tr> <td>fraudulent</td> <td>< 10,000</td> </tr> </tbody> </table>	Transaction Type	Count	legitimate	< 10,000	fraudulent	< 10,000
Transaction Type	Count							
legitimate	< 10,000							
fraudulent	< 10,000							

Resampling method	Description	Target class distribution after resampling						
Oversampling (SMOTE)	<p>Generate new synthetic fraudulent transactions until the number of fraudulent transactions is ca. equal to the number of legitimate transactions:</p> <ol style="list-style-type: none"> Select one of the fraudulent transactions in the training data randomly Select one of its n nearest neighbors in the same fraudulent class randomly Select a random point between the existing fraudulent transaction and its nearest neighbor 	<ul style="list-style-type: none"> Original data in yellow New synthetic data in light patterned yellow  <table border="1"> <thead> <tr> <th>Transaction Type</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>legitimate</td> <td>~200,000</td> </tr> <tr> <td>fraudulent</td> <td>~200,000</td> </tr> </tbody> </table>	Transaction Type	Count	legitimate	~200,000	fraudulent	~200,000
Transaction Type	Count							
legitimate	~200,000							
fraudulent	~200,000							

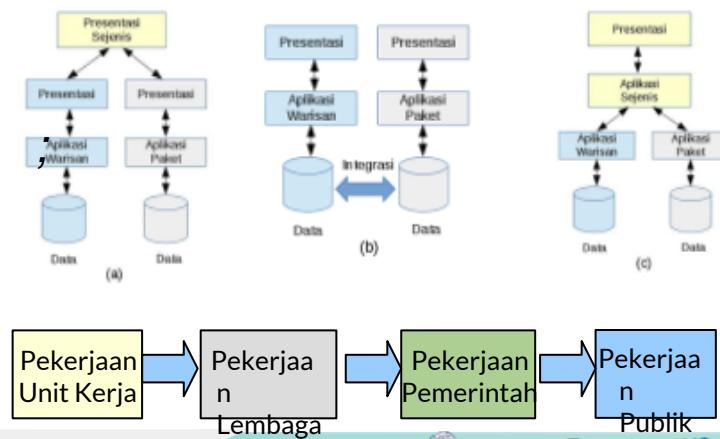


Keberadaan data dari berbagai sumber (stakeholder)



Sistem informasi di masing-masing organisasi tidak bisa bertukar data/informasi pada lingkungan heterogen
 Interoperabilitas data akan mengefisienkan kerja serta dapat melakukan **prediksi dan analisis berbasiskan AI**
 Mendukung knowledge discovery dan decision making

- **Integrasi Presentasi.** User interface yang menyediakan akses pada suatu aplikasi. kinerja, persepsi, dan tidak adanya interkoneksi antara aplikasi dan data.
- **Integrasi Data.** Dilakukan langsung pada basis data atau struktur data. Jika terjadi perubahan model data, maka integrasinya perlu direvisi atau dilakukan ulang.
- **Integrasi Fungsional** Proses integrasi dilakukan pada level logika bisnis pada beberapa aplikasi.



Kelengkapan Data sesuai Tujuan



Data Perencanaan

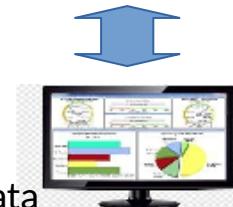
Data Pelaksanaan

Data Pengawasan

Data Penindakan

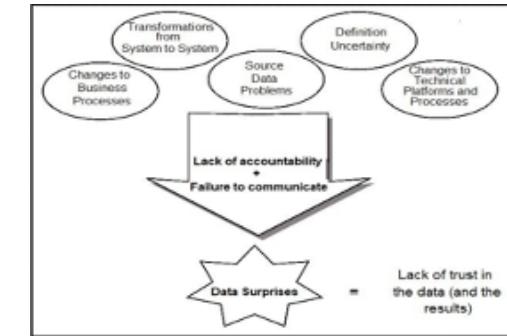
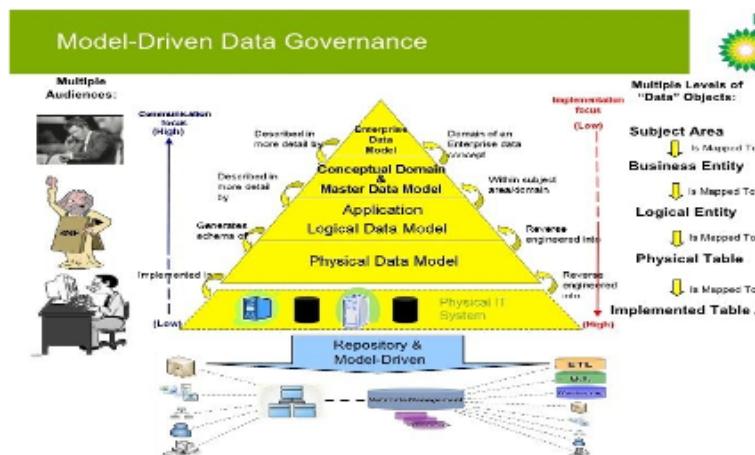
Penggunaan data/fungsi bersama -interoperabilitas

- Menaikkan kualitas data di lingkungan organisasi
- Konsistensi data dan update dijaga
- Mengupayakan agar memiliki skema yang sama ataupun pemetaan skema yang terbuka → skema data diketahui umum
- Mengupayakan data referensi sama → data referensi diketahui



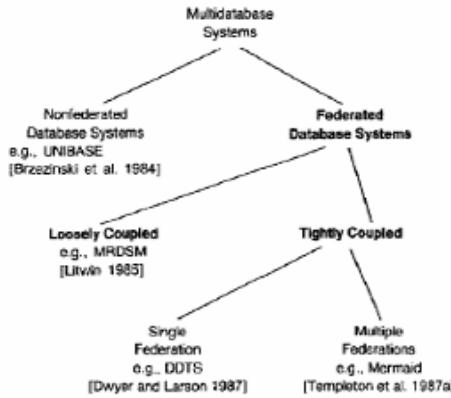


Kualitas Data bergantung Governance

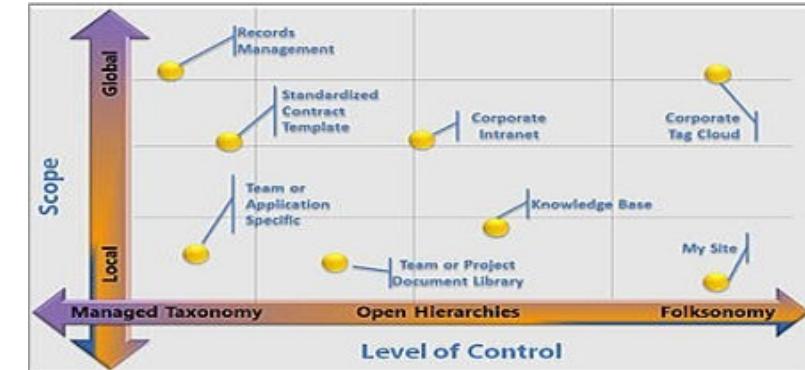




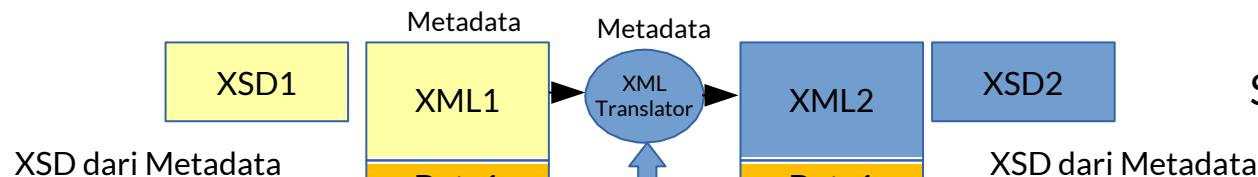
Federated Database



- Tidak mungkin memaksa setiap pihak “menyerahkan” datanya
- Setiap pihak memiliki teknologi dan sistem masing-masing
- Transparency
- Heterogeneity
- Functionality
- Autonomy of underlying federated sources,
- Extensibility & Openness
- Optimized performance



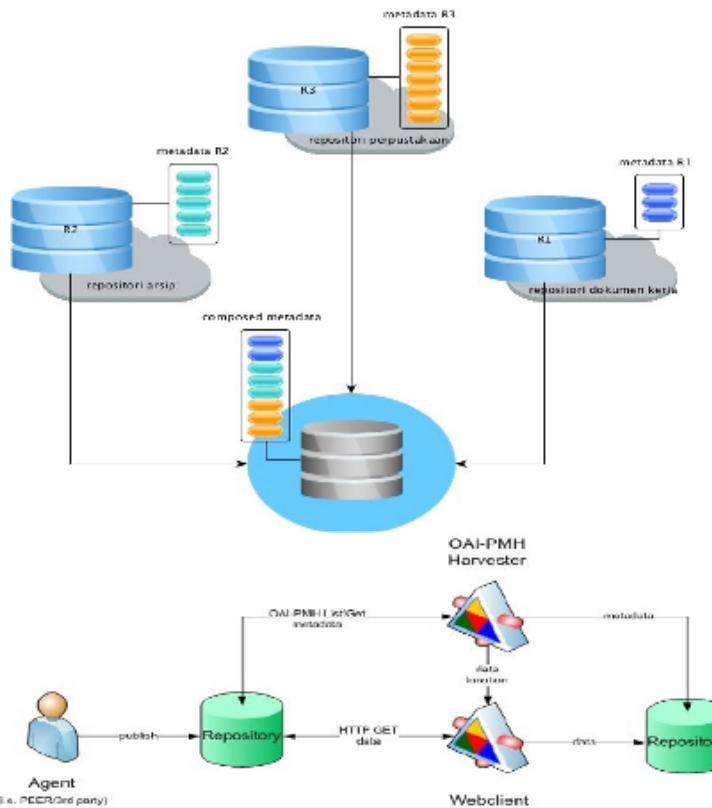
Sistem A



DocID	FieldA	FieldB	FieldC

ID	FieldX	FieldY

Ontologi dan Database



DATABASE RELATIONAL

Close World Assumption (CWA)
Fokus pada data

Adanya Constraint untuk mencapai data integritas, namun mungkin menyembunyikan makna

Tidak menggunakan hirarki IS

Skema lebih sederhana, belum tentu dapat digunakan kembali

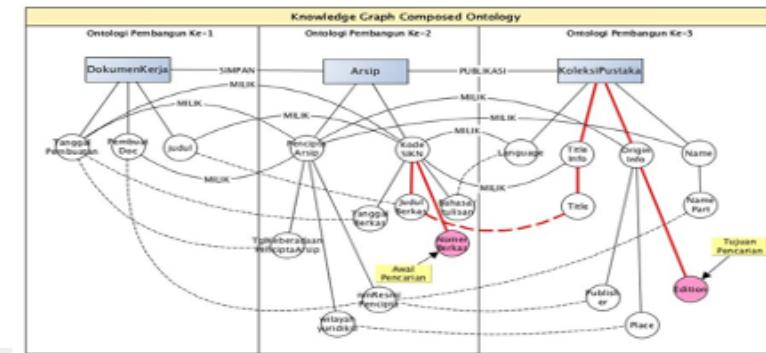
ONTOLOGI

Open World Assumption (OWA),
Fokus pada makna

Adanya Ontology axioms untuk menspesifikasi makna, dapat digunakan untuk pencapaian integritas

Hirarki ISA merupakan backbone

Skema lebih kompleks, dapat digunakan kembali



Pemilihan (Seleksi Fitur) Data

	K1	K2	K3	K4	K5	K6
R1						
R2						
R3						
R4						

Fitur:
 Kolom yang dipilih
 Untuk sesuai tujuan

Name of the statistical features	Formula/description
Standard error	$\sqrt{\frac{1}{n-2} \left[\sum (y - \bar{y})^2 - \frac{\sum (x_i - \bar{x})(y_j - \bar{y}) ^2}{\sum (x_i - \bar{x})^2} \right]}$
Standard deviation	$\sqrt{\frac{n \sum x^2 - (\sum x)^2}{4(n-1)}}$
Sample variance	$\sqrt{\frac{n \sum x^2 - (\sum x)^2}{n(n-1)}}$
Kurtosis	$\left\{ \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum \left(\frac{x_i - \bar{x}}{s_x} \right)^4 \right\} - \frac{3(n-1)^2}{(n-2)(n-3)}$
Skewness	$\frac{n-1}{(n-2)} \sum \left(\frac{x_i - \bar{x}}{s_x} \right)^3$
Maximum value	Maximum signal point value in a given signal.
Minimum value	Minimum signal point value in a given signal.
Range	Difference in maximum and minimum signal point values for a given signal.
Sum	Sum of all feature values for each sample.
Mean	The arithmetic average of a set of values or distribution.
Median	Middle value separating the greater and lesser halves of a data set.
Mode	A statistical term that refers to the most frequently occurring number found in a set of numbers. (i.e.) The

- Setelah menentukan sampling atas data yang akan diambil nanti, selanjutnya adalah melakukan seleksi **fitur** (feature selection) atas data sampling tsb --> Memilih Kolom/Atribut/Variabel yang akan diolah lebih lanjut Terminologi **fitur** di Data Science atau Machine Learning adalah Kolom/Atribut/Variabel yang dianggap & dihitung sebagai prioritas (sedikit berbeda dengan terminologi fitur di Statistika)
- Seleksi fitur merupakan konsep inti dalam ML yang berdampak besar bagi kinerja model prediksi,
- Fitur data yang tidak/sebagian saja relevan dampak berdampak negatif thdp kinerja model
- Definisi Seleksi Fitur: proses otomatis atau manual memilih fitur data yang **paling berkontribusi** thdp variabel prediksi
- atau output yang diinginkan.

Validasi Data

- Verifikasi vs ValiditasVerifikasi: Benar vs Salah (sesuai prosedur)
 - Validasi: Kuat vs Lemah (sesuai kenyataan)
- Validasi merupakan tahapan kritikal yang sering diabaikan DS-tist pemula, karena memeriksa, diantaranya sbb:
 - Tipe Data (mis. integer, float, string) Range Data
 - Uniqueness (mis. Kode Pos)
 - Consisten expression (mis. Jalan, Jl., Jln.)
 - Format Data (mis. utk tgl "YYYY-MM-DD" VS "DD-MM-YYYY.") → tmt (terhitung mulai tanggal)
 - Nilai Null/Missing Values
 - Misspelling/Type
 - Invalid Data (gender: L/P: L; Laki-laki; P: Pria/Perempuan?)
- Teknik Validasi Data dan Model:
 - Akurasi Kelengkapan
 - Konsistensi Ketepatan
 - Waktu Kepercayaan
 - Nilai Tambah Penafsiran
 - Kemudahan Akses

Pandas: DataFrame

Syntax – Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = [1, 2, 3])
Specify values for each column.
```

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
Specify values for each row.
```

	a	b	c
m			
n			
d	1	4	7
e	2	5	8
f	3	6	9
g	4	7	10
h	5	8	11
i	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = pd.MultiIndex.from_tuples(
        [('d',1),('d',2),('e',2)],
        names=['n', 'v']))
Create DataFrame with a MultiIndex
```

Reshaping Data – Change the layout of a data set



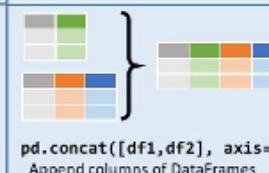
pd.melt(df)
 Gather columns into rows.



df.pivot(columns='var', values='val')
 Spread rows into columns.



pd.concat([df1, df2])
 Append rows of DataFrames



pd.concat([df1, df2], axis=1)
 Append columns of DataFrames

df.sort_values('mpg')
 Order rows by values of a column (low to high).

df.sort_values('mpg', ascending=False)
 Order rows by values of a column (high to low).

df.rename(columns = {'y':'year'})
 Rename the columns of a DataFrame

df.sort_index()
 Sort the index of a DataFrame

df.reset_index()
 Reset index of DataFrame to row numbers, moving index to columns.

df.drop(columns=['Length', 'Height'])
 Drop columns from DataFrame

Subset Observations (Rows)



```
df[df.Length > 7]
Extract rows that meet logical
criteria.
df.drop_duplicates()
Remove duplicate rows (only
considers columns).
df.head(n)
Select first n rows.
df.tail(n)
Select last n rows.
```

```
df.sample(frac=0.5)
Randomly select fraction of rows.
df.sample(n=10)
Randomly select n rows.
df.iloc[10:20]
Select rows by position.
df.nlargest(n, 'value')
Select and order top n entries.
df.nsmallest(n, 'value')
Select and order bottom n entries.
```

Subset Variables (Columns)



```
df[['width', 'length', 'species']]
Select multiple columns with specific names.
df['width'] or df.width
Select single column with specific name.
df.filter(regex='regEx')
Select columns whose name matches regular expression regex.
```



Hands-on Data Cleaning

- Data cleaning atas data berantakan (messy data), seperti:
 - missing value,
 - format tidak konsisten
 - record tidak berbentuk baik (malformed record)
 - outlier yang berlebihan
- Lingkup hands-on:
 - Membuang kolom-kolom tidak penting dalam suatu DataFrame
 - Mengubah indeks di DataFrame
 - Membersihkan kolom dengan metode `.str()`
 - Membersihkan semua dataset dengan fungsi `DataFrame.applymap()`
 - Merubah nama kolom sehingga kolom lebih mudah dikenali
 - Melewaskan baris-baris tidak penting dalam file CSV

Hands-on Data Cleaning: Datasets

- File CSV tentang “Daftar Buku dari British Library”, nama file “BL-Flickr-Images-Book.csv”, link:
- <https://github.com/realpython/python-data-cleaning/blob/master/Datasets/BL-Flickr- Images-Book.csv>
- File teks tentang “Kota lokasi Sekolah Tinggi di US”, nama file “university_towns.txt”, link: https://github.com/realpython/python-data- cleaning/blob/master/Datasets/university_towns.txt
- File CSV tentang “Partisipasi Semua Negara di Olimpiade Musim Dingin dan Musim Panas”, nama file “olympics.csv”, link: <https://github.com/realpython/python-data-cleaning/blob/master/Datasets/olympics.csv>



Hands-on Data Cleaning: Import modul

Diasumsikan peserta sudah memahami library Pandas dan NumPy (lihat di pertemuan sebelumnya) termasuk Pandas workhouse Series dan objek DataFrame

1. Import modul yang dibutuhkan

```
import pandas as pd  
import numpy as np
```

Jika ingin melihat statistik dasar pada DataFrame di Pandas dengan fungsi .describe():

```
df.describe()
```

Hands-on Data Cleaning: Membuang (drop) Kolom

- Membuang Kolom pada DataFrame
- Sering ditemukan bbrp kategori data tidak terlalu berguna di dataset, misal untuk menganalisis IPK mahasiswa , data nama orangtua, alamat adalah data tidak penting
- Pandas menyediakan fungsi untuk membuang (drop) kolom-kolom yang tidak diinginkan dengan fungsi `drop()`.
 1. Buat DataFrame di luar file CSV . Dalam contoh berikut kita lewatkan path relatif ke `pd.read.csv`, yaitu seluruh dataset berada di nama folder Datasets di direktori kerja

Hands-on Data Cleaning: Membuang (drop) Kolom

```
df = pd.read_csv("C:/Users/Bayu/Documents/DTS 2021/Datasets/BL-Flickr-Images-Book.csv")
```

```
df.head()
```

	Identifier	Edition Statement	Place of Publication	Date of Publication	Publisher	Title	Author	Contributors	Corporate Author	Corporate Contributors	Former owner	Engraver	Issuance type
0	206	NaN	London	1879 [1878]	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A.	A. A.	FORBES, Walter.	NaN	NaN	NaN	NaN	monographic http
1	216	NaN	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...]	A., A. A.	BLAZE DE BURY, Marie Pauline Rose - Baroness	NaN	NaN	NaN	NaN	monographic http
2	218	NaN	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of 'All for Gr...	A., A. A.	BLAZE DE BURY, Marie Pauline Rose - Baroness	NaN	NaN	NaN	NaN	monographic http
3	472	NaN	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A. E. S.	Appleyard, Ernest Silvanus.	NaN	NaN	NaN	NaN	monographic http

df.head() : berfungsi untuk menampilkan baris awal dari DataFrame, secara default bila tdk diberi parameter akan menampilkan 5 baris data

Hands-on Data Cleaning: Membuang (drop) Kolom

- Melihat pada lima entri pertama dengan perintah `head()`.
- Dapat dilihat bahwa beberapa kolom memberikan informasi tambahan yang akan membantu perpustakaan tetapi tidak terlalu deskriptif tentang buku itu sendiri:

`Edition Statement`, `Corporate Author`, `Corporate Contributors`, `Former owner`,
`Engraver`, `Issuance type` and `Shelfmarks`.

Kita drop kolom-kolom tsb dengan perintah:

```
to_drop = ['Edition Statement',
           'Corporate Author',
           'Corporate Contributors',
           'Former owner',
           'Engraver',
           'Contributors',
           'Issuance type',
           'Shelfmarks']
```

```
df.drop(to_drop, inplace=True, axis=1)
```

Kita definisikan daftar (list) nama dari semua kolom yang ingin kita drop.
Kemudian jalankan perintah fungsi `drop()`.

dengan parameter `inplace` bernilai `True` dan parameter `axis` bernilai `1`, di mana `1` adalah angka sumbu (`0` untuk baris dan `1` untuk kolom.)

Hands-on Data Cleaning: Membuang (drop) Kolom

- Inspeksi ulang DataFrame, kolom yang tidak diinginkan sudah dibuang

```
df.head()
```

	Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
0	206	London	1879 [1878]	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	http://www.flickr.com/photos/britishlibrary/ta...
1	216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...]	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
2	218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of "All for Gr...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
3	472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...
4	480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...]	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...

Hands-on Data Cleaning: Membuang (drop) Kolom

- Alternatif utk membuang kolom, dengan meneruskannya langsung ke parameter columns daripada memisahkan label-label yang mau dibuang:

```
df.drop(columns=to_drop, inplace=True)
```

inplace=**True** adalah perintah menimpa kolom dan menyimpan kolom/fitur yang dimanipulasi (dlm hal ini di drop)

- Sintak ini lebih intuitif dan mudah dibaca dibanding sintak sebelumnya

Handling Missing Data

```
df.dropna()
```

Drop rows with any column having NA/null data.

```
df.fillna(value)
```

Replace all NA/null data with value.

```
df.drop(to_drop, inplace=True, axis=1)
```



Hands-on Data Cleaning: Mengubah Indeks di DataFrame

- Index dalam Pandas memperluas fungsionalitas array NumPy untuk memungkinkan pemotongan (slicing) dan pelabelan yang lebih fleksibel. Dalam banyak kasus, akan sangat membantu jika menggunakan field pengenal data yang bernilai unik sebagai indeksnya.
- Sebagai contoh, dengan dataset di slide sebelumnya, praktiknya saat pustakawan mencari record, biasanya akan memasukan identifier unik suatu buku:

```
df['Identifier'].is_unique
```

True

df['Identifier']: slicing/seleksi kolom/field yang akan di eksekusi.

.is_unique: function untuk mengecek nilai unik

Hands-on Data Cleaning: Mengubah Indeks di DataFrame

- Gantikan indeks yang ada pada kolom ini menggunakan `set_index`

`.set_index()`: function untuk merubah index dengan diikuti parameter kolom yang akan dipilih untuk dijadikan index

```
df = df.set_index('Identifier')  
df.head()
```

	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
Identifier						
206	London	1879 [1878]	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	http://www.flickr.com/photos/britishlibrary/ta...
216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of "All for Gr...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...
480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...

Technical Detail: Unlike primary keys in SQL, a Pandas Index doesn't make any guarantee of being unique, although many indexing and merging operations will notice a speedup in runtime if it is.

Hands-on Data Cleaning: Mengubah Indeks di DataFrame

- Setiap records dapat diakses dengan cara yang mudah dengan `loc []`. Cara ini digunakan untuk *label-based indexing*, yaitu memberi label suatu baris atau kolom tanpa memperhatikan posisi/lokasinya.

```
df.loc[206]
```

```
Place of Publication          London
Date of Publication          1879 [1878]
Publisher                    S. Tinsley & Co.
Title                        Walter Forbes. [A novel.] By A. A
Author                       A. A.
Flickr URL                  http://www.flickr.com/photos/britishlibrary/ta...
Name: 206, dtype: object
```

Dengan kata lain, 206 adalah label pertama dari indeks. Utk mengakses berdasarkan posisinya, gunakan `df.iloc []`

loc: untuk seleksi dengan menggunakan label/bilangan bulat
iloc: untuk seleksi dengan menggunakan bilangan bulat
contoh penggunaan lain:
`iin.iloc[::3,:].head(10)`: untuk Memilih baris kelipatan 3, dengan semua kolom dan menampilkan 10 data pertama.

Hands-on Data Cleaning: Mengubah Indeks di DataFrame

- Pada slide sebelumnya, Indeks yang digunakan adalah `RangeIndex: integer` mulai dari 0, analog dengan range di Python. Dengan meneruskan nama kolom ke `set_index`, maka indeks telah diubah ke nilai dalam Identifier.
- Diperhatikan pada langkah sebelumnya bahwa telah dilakukan penetapan kembali variabel ke objek yang dikembalikan oleh metode dengan `df = df.set_index(...)`. Ini karena, secara default, metode mengembalikan salinan objek yang dimodifikasi dan tidak membuat perubahan secara langsung ke objek. Hal ini dapat dihindari dengan mengatur parameter `inplace`:

```
df.set_index('Identifier', inplace=True)
```



Hands-on Data Cleaning: Merapikan *Fields* dalam Data

- Slide sebelumnya telah dibuang beberapa kolom tidak penting dan diubah indeks pada DataFrame hingga menjadi lebih masuk akal.
- Selanjutnya, akan dibersihkan kolom tertentu dan mengubah menjadi bentuk/format yang seragam hingga dataset lebih mudah dipahami dan memastikan konsistensi. Dalam slide berikutnya akan dibersihkan Date of Publication dan Place of Publication.
- Dalam inspeksi, semua tipe data saat ini adalah objek `dtype` yang analog dengan `str` di native Python



Hands-on Data Cleaning: Merapikan *Fields* dalam Data

- Cara ini dilakukan sebagai rangkuman saat setiap field tidak dapat dirapikan sebagai data numerik atau data kategorik dan data yang digunakan cukup “kotor” atau “berantakan”.

```
df.dtypes.value_counts()
```

```
object    6
dtype: int64
```

Hands-on Data Cleaning: Merapikan *Fields* dalam Data

Satu kolom yang masuk akal untuk menerapkan nilai numerik adalah tanggal publikasi sehingga kita dapat melakukan perhitungan di awal:

Buku tertentu hanya memiliki satu tanggal publikasi.
Oleh karena itu perlu dilakukan hal berikut:

- Hilangkan tanggal lain dalam kurung siku,
1879 [1878]
- Konversi rentang tanggal ke “start date”, 1860–
63; 1839, 38–54
- Hilangkan tanggal yang tidak jelas dan gantikan
dengan NaN NumPy, [1879?] -> NaN
- Konversi string nan ke nilai NaN NumPy

```
df.loc[1905:, 'Date of Publication'].head(10)
```

Identifier	Date of Publication
1905	1888
1929	1839, 38-54
2836	1897
2854	1865
2956	1860-63
2957	1873
3017	1866
3131	1899
4598	1814
4884	1820

```
Name: Date of Publication, dtype: object
```

df.loc[1905:, ‘Nama Field’]:
digunakan untuk mengakses index mulai dari
index 1905 dengan output hanya pada field
tanggal publikasi

.head(10) : function untuk menampilkan baris
awal DataFrame dengan parameter hingga index ke
10 atau 10 baris data



Hands-on Data Cleaning: Merapikan *Fields* dalam Data

- Mensintesis pola-pola ini, manfaatkan ekspresi reguler (Regex) tunggal untuk mengekstrak tahun publikasi.

```
regex = r'^(\d{4})'
```

- perintah \d mewakili sembarang digit dan { 4 } mengulangi aturan (rule) sebanyak empat kali. Karakter ^ sesuai dengan awal string, dan tanda dalam kurung () menunjukkan *capturing group* yang memberikan sinyal ke Pandas bahwa akan dilakukan ekstraksi bagian Regex tersebut.



Hands-on Data Cleaning: Regex di Pandas

regex (Regular Expressions) Examples	
'\.'	Matches strings containing a period ''
'Length\$'	Matches strings ending with word 'Length'
'^Sepal'	Matches strings beginning with the word 'Sepal'
'^x[1-5]\$'	Matches strings beginning with 'x' and ending with 1,2,3,4,5
'^(?!Species\$).*''	Matches strings except the string 'Species'

`df.loc[:, 'x2':'x4']`

Select all columns between x2 and x4 (inclusive).

`df.iloc[:, [1, 2, 5]]`

Select columns in positions 1, 2 and 5 (first column is 0).

`df.loc[df['a'] > 10, ['a', 'c']]`

Select rows meeting logical condition, and only the specific columns .

Hands-on Data Cleaning: Merapikan *Fields* dalam Data

- Coba jalankan regex di dataset

```
extr = df['Date of Publication'].str.extract(r'^(\d{4})', expand=False)  
extr.head()
```

Identifier
206 1879
216 1868
218 1869
472 1851
480 1857
Name: Date of Publication, dtype: object

Further Reading: Not familiar with regex? You can [inspect the expression above at regex101.com](#) and learn all about regular expressions with Regular Expressions: [Regexes in Python](#).

Mengekstrak data untuk setiap string subjek hasil tangkapan variabel regex dari kolom Date of Publication

expand=False: Jika Benar, kembalikan DataFrame dengan satu kolom per grup tangkapan. Jika Salah, kembalikan Seri/Indeks jika ada satu grup tangkapan atau DataFrame jika ada beberapa grup tangkapan.

Hands-on Data Cleaning: Merapikan *Fields* dalam Data

- Secara teknis, kolom tsb masih memiliki dtype = object, namun dengan mudah kita dapatkan versi numeriknya dengan perintah pd.to_numeric

```
df['Date of Publication'] = pd.to_numeric(extr)
df['Date of Publication'].dtype
dtype('float64')
```

- Ini menghasilkan sekitar 1/10 nilai yang hilang, cost yang cukup kecil dampaknya untuk saat ini karena dapat melakukan perhitungan pada nilai valid yang tersisa:

```
df['Date of Publication'].isnull().sum() / len(df)
0.11717147339205986
```

Hands-on Data Cleaning: Membersihkan Kolom dengan Kombinasi metode str dengan NumPy

- Slide sebelumnya dibahas penggunaan `df['Date of Publication'].str`. Atribut ini adalah cara akses cepat operasi string di Pandas yang menyerupai operasi pada native Python atau mengkompilasi regex seperti `.split()`, `.replace()`, dan `.capitalize()`.
- Utk membersihkan field `Place of Publication`, kombinasikan metode `str` di Pandas dengan fungsi `np.where` di NumPy yang mirip dengan bentuk vektor dari makro `IF()` di Excel, dengan sintak berikut:

```
np.where(condition, then, else)
```



Hands-on Data Cleaning: Membersihkan Kolom dengan Kombinasi metode str dengan NumPy

```
np.where(condition, then, else)
```

- condition mirip dengan objek array atau Boolean .then adalah nilai yang digunakan jika condition mengevaluasi menjadi True, dan else untuk mengevaluasi nilai selainnya.
- .where membawa tiap elemen dalam objek digunakan untuk condition dan memeriksa elemen tertentu menjadi True dalam konteks kondisi dan mengembalikan ndarray terdiri dari then atau else, tergantung pada prakteknya.



Hands-on Data Cleaning: Membersihkan Kolom dengan Kombinasi metode `str` dengan NumPy

- Dapat juga dituliskan dalam bersarang (nested) menjadi pernyataan *If-Then*, memungkinkan menghitung nilai berdasarkan kondisi berganda:

Python

>>>

```
>>> np.where(condition1, x1,  
           np.where(condition2, x2,  
           np.where(condition3, x3, ...)))
```

- Kemudian, dapat digunakan dua fungsi tsb untuk membersihkan field Place of Publication karena kolom tsb memiliki objek string. Berikut adalah isi dari kolom:

Hands-on Data Cleaning: Membersihkan Kolom dengan Kombinasi metode str dengan NumPy

```
df['Place of Publication'].head(10)
```

```
Identifier
206                               London
216      London; Virtue & Yorston
218                               London
472                               London
480                               London
481                               London
519                               London
667      pp. 40. G. Bryan & Co: Oxford, 1898
874                               London]
1143                               London
Name: Place of Publication, dtype: object
```

- Dilihat pada hasil di atas, field place of publication masih ada informasi yang tidak penting. Jika dilihat lebih teliti, kasus ini untuk beberapa baris yang place of publication -nya di “London” dan “Oxford”



Hands-on Data Cleaning: Membersihkan Kolom dengan Kombinasi metode str dengan NumPy

```
df.loc[4157862]
```

```
Place of Publication      Newcastle-upon-Tyne
Date of Publication       1867.0
Publisher                 T. Fordyce
Title                     Local Records; or, Historical Register of rema...
Author                    FORDYCE, T. - Printer, of Newcastle-upon-Tyne
Flickr URL                http://www.flickr.com/photos/britishlibrary/ta...
Name: 4157862, dtype: object
```

```
df.loc[4159587]
```

```
Place of Publication      Newcastle upon Tyne
Date of Publication       1834.0
Publisher                 Mackenzie & Dent
Title                     An historical, topographical and descriptive v...
Author                    Mackenzie, E. (Eneas)
Flickr URL                http://www.flickr.com/photos/britishlibrary/ta...
Name: 4159587, dtype: object
```

- Pada dua entri di samping, dua buku diterbitkan di tempat yang sama (newcastle upon tyne) namun salah satunya memiliki tanda hubung (-)
- Untuk membersihkan kolom ini dalam sekali jalan, gunakan `str.contains()` untuk mendapatkan Boolean mask.

```
pub = df['Place of Publication']
london = pub.str.contains('London')
london[:5]
```

```
Identifier
206   True
216   True
218   True
472   True
480   True
Name: Place of Publication, dtype: bool
```

```
oxford = pub.str.contains('Oxford')
```

Hands-on Data Cleaning: Membersihkan Kolom dengan Kombinasi metode str dengan NumPy

- Kombinasikan dengan np.where:

```
df['Place of Publication'] = np.where(london, 'London',
                                         np.where(oxford, 'Oxford',
                                                 pub.str.replace('-', ' ')))
df['Place of Publication'].head()

Identifier
206    London
216    London
218    London
472    London
480    London
Name: Place of Publication, dtype: object
```

- Di sini, fungsi np.where berbentuk struktur nested, dimana condition berbentuk Series dari Boolean dengan str.contains(). Metode contains() bekerja mirip dengan keyword in yang digunakan untuk mencari kejadian suatu entitas dalam kondisi pengulangan iterasi (atau substring dalam suatu string)



Hands-on Data Cleaning: Membersihkan Kolom dengan Kombinasi metode str dengan NumPy

- Pergantian tanda hubung (hyphen) dengan spasi dengan `str.replace()` dan re-assign ke kolom dalam DataFrame.
- Walau pada kenyataan masih banyak dataset ini (kolom dan baris) yang “kotor” , namun dalam contoh di sini hanya dibahas pada dua kolom

Hands-on Data Cleaning: Membersihkan Kolom dengan Kombinasi metode str dengan NumPy

- Coba periksa kembali untuk lima entri pertama, hasilnya akan lebih rapi dan “bersih” dibandingkan dataset awal sebelum dilakukan *cleaning data*.

```
df.head()
```

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
206	London	1879.0	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A.	A. A.	http://www.flickr.com/photos/britishlibrary/ta...
216	London	1868.0	Virtue & Co.	All for Greed. [A novel. The dedication signed...]	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
218	London	1869.0	Bradbury, Evans & Co.	Love the Avenger. By the author of "All for Gr...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...
472	London	1851.0	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...
480	London	1857.0	Wertheim & MacIntosh	[The World in which I live, and my place in it...]	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...

Note: At this point, Place of Publication would be a good candidate for conversion to a Categorical dtype, because we can encode the fairly small unique set of cities with integers. (The memory usage of a Categorical is proportional to the number of categories plus the length of the data; an object dtype is a constant times the length of the data.)



Hands-on Data Cleaning: Membersihkan Seluruh Dataset dengan Fungsi `applymap`

- Pada situasi tertentu, data berantakan alias “kotor” tidak hanya berlaku di sebagian kolom atau baris (record) tapi menyebar ke banyak bagian dataset. Cara berikut dapat diterapkan untuk semua cell atau elemen di **DataFrame** (dataset).
- Metode `.applymap()` dapat diterapkan, dimana similar dengan fungsi built-in yaitu fungsi `map()`.

Hands-on Data Cleaning: Membersihkan seluruh Dataset

- Terapkan fungsi `applymap()` pada file “university_towns.txt”:

Shell

```
$ head Datasets/univerisity_towns.txt
Alabama[edit]
Auburn (Auburn University)[1]
Florence (University of North Alabama)
Jacksonville (Jacksonville State University)[2]
Livingston (University of West Alabama)[2]
Montevallo (University of Montevallo)[2]
Troy (Troy University)[2]
Tuscaloosa (University of Alabama, Stillman College, Shelton State)[3][4]
Tuskegee (Tuskegee University)[5]
Alask[edit]
```

Perintah shell

- Dapat dilihat di atas, bahwa nama negara bagian(state) diikuti dengan kota asal universitas:
StateA TownA1 TownA2 StateB TownB1 TownB2....
dan memiliki substring “[edit]”

Hands-on Data Cleaning: Membersihkan seluruh Dataset

- Kita dapat memanfaatkan pola ini dengan membuat list of (state, city) tuples dan *wrapping* daftar (list) dalam DataFrame.

```
university_towns = []
with open("C:/Users/Bayu/Documents/DTS 2021/Datasets/university_towns.txt") as file:
    for line in file:
        if '[edit]' in line:
            # Remember this `state` until the next is found
            state = line
        else:
            # Otherwise, we have a city; keep `state` as last-seen
            university_towns.append((state, line))

university_towns[:5]
```

```
[('Alabama[edit]\n', 'Auburn (Auburn University)[1]\n'),
 ('Alabama[edit]\n', 'Florence (University of North Alabama)\n'),
 ('Alabama[edit]\n', 'Jacksonville (Jacksonville State University)[2]\n'),
 ('Alabama[edit]\n', 'Livingston (University of West Alabama)[2]\n'),
 ('Alabama[edit]\n', 'Montevallo (University of Montevallo)[2]\n')]
```

- Kita dapat membungkus (wrap) daftar ini dalam DataFrame dan mengatur kolom sebagai "State" and "RegionName".
- Pandas akan mengambil setiap elemen dalam daftar dan mengatur "State" ke nilai kiri dan "RegionName" ke nilai kanan.
- Hasilnya adalah DataFrame sbb:

```
towns_df = pd.DataFrame(university_towns,
                        columns=['State', 'RegionName'])

towns_df.head()
```

	State	RegionName
0	Alabama[edit]\n	Auburn (Auburn University)[1]\n
1	Alabama[edit]\n	Florence (University of North Alabama)\n
2	Alabama[edit]\n	Jacksonville (Jacksonville State University)[2]\n
3	Alabama[edit]\n	Livingston (University of West Alabama)[2]\n
4	Alabama[edit]\n	Montevallo (University of Montevallo)[2]\n



Hands-on Data Cleaning: Membersihkan seluruh Dataset

- Pandas, mempermudah dalam pembersihan string dengan hanya membutuhkan nama state dan nama town dan dapat membuang lainnya. Selain dapat kembali menggunakan metode `.str()` di Pandas, dapat juga menggunakan `applymap()` untuk memetakan setiap elemen di DataFrame
- Perhatikan kasus sederhana pada contoh DataFrame berikut:

	0	1
0	mock	Dataset
1	python	pandas
2	real	python
3	numpy	clean

- Pada contoh di atas, setiap sel (“Mock”, “Dataset”, “Python”, “Real”, dll) adalah elemen. Oleh karena itu perintah `applymap()` akan menerapkan fungsi ke setiap elemen secara independen. Mari kita definisikan fungsi tsb:



Hands-on Data Cleaning: Membersihkan seluruh Dataset

- Fungsinya didefinisikan

```
def get_citiesstate(item):
    if ' (' in item:
        return item[:item.find(' (')]
    elif '[' in item:
        return item[:item.find('[')]
    else:
        return item
```

- applymap () di Pandas hanya butuh satu parameter , yaitu fungsi yang diterapkan ke setiap elemen:

```
towns_df = towns_df.applymap(get_citiesstate)
```

- Pertama, definisikan fungsi Python yang mengambil setiap elemen dari DataFrame sebagai parameternya. Di dalam fungsi, pengecekan dilakukan utk menentukan apakah ada elemen atau tidak.

Hands-on Data Cleaning: Membersihkan data dengan applymap()

- Tergantung pada pengecekan, nilai dikembalikan berdasarkan fungsi.
- Lalu, fungsi applymap() dipanggil pada objek yg ada. Sehingga kita dapatkan DataFrame yang relatif lebih rapi

```
towns_df.head()
```

	State	RegionName
0	Alabama	Auburn
1	Alabama	Florence
2	Alabama	Jacksonville
3	Alabama	Livingston
4	Alabama	Montevallo

Technical Detail: While it is a convenient and versatile method, .applymap can have significant runtime for larger datasets, because it maps a Python callable to each individual element. In some cases, it can be more efficient to do vectorized operations that utilize Cython or NumPY (which, in turn, makes calls in C) under the hood.

Hands-on Data Cleaning: Mengganti Nama Kolom

- Seringkali dalam dataset yang dimiliki ada nama kolom yang sulit untuk dipahami atau informasi yang tidak penting dalam beberapa baris awal/akhir, misal definisi istilah atau footnotes.
- Oleh karena itu dapat dilakukan penggantian nama dan melewatkannya beberapa baris sehingga bisa dilakukan analisis informasi dari baris yang benar atau dapat dipahami.
- Kita akan lakukan untuk lima baris awal dataset “olympic.csv”:

Shell

```
$ head -n 5 Datasets/olympics.csv
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
,? Summer,01 !,02 !,03 !,Total,? Winter,01 !,02 !,03 !,Total,? Games,01 !,02 !,03 !,Co
Afghanistan (AFG),13,0,0,2,2,0,0,0,0,13,0,0,2,2
Algeria (ALG),12,5,2,8,15,3,0,0,0,0,15,5,2,8,15
Argentina (ARG),23,18,24,28,70,18,0,0,0,0,41,18,24,28,70
```

Hands-on Data Cleaning: Mengganti Nama Kolom

- Kemudian, baca dalam DataFrame di Pandas:

```
olympics_df = pd.read_csv("C:/Users/Bayu/Documents/DTS 2021/Datasets/olympics.csv")
olympics_df.head()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	NaN	? Summer	01!	02!	03!	Total	? Winter	01!	02!	03!	Total	? Games	01!	02!	03!	Combined total
1	Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	0	2	2
2	Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	2	8	15
3	Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	24	28	70
4	Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	2	9	12

- Hasilnya berantakan! Kolom adalah bentuk string integer indeks 0. Baris yang harusnya sebagai header pada `olympics_df.iloc[0]`. Hal ini terjadi karena file CSV mulai dengan 0, 1, 2, ..., 15.
- Dan, jika kita ke sumber dataset ini, akan terlihat NaN yang ada harusnya berisikan “Country” dan “Summer” maksudnya adalah “Summer Games” dan “01!” harusnya adalah “Gold”, dll.



Hands-on Data Cleaning: Mengganti Nama Kolom

Oleh karena itu, hal berikut yang perlu dilakukan:

- Melewatkkan (skip) satu baris dan atur header sebagai baris pertama (indeks-0)
- Mengganti Nama Kolom
- Melewatkkan baris dan atur header dapat dilakukan pada saat membaca file CSV dengan mempassing beberapa parameter ke fungsi `read_csv()`.
- Fungsi `read_csv()` memerlukan banyak parameter opsional, namun utk kasus ini hanya diperlukan satu (header) yang dihilangkan pada baris ke-0

Hands-on Data Cleaning: Mengganti Nama Kolom

- Hasil fungsi `read_csv()` dan menghilangkan satu baris (header)

```
olympics_df = pd.read_csv("C:/Users/Bayu/Documents/DTS 2021/Datasets/olympics.csv", header=1)
olympics_df.head()
```

	Unnamed: 0	Summer	01!	02!	03!	Total	Winter	01!1	02!1	03!1	Total.1	?Games	01!2	02!2	03!2	Combined total
0	Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	0	2	2
1	Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	2	8	15
2	Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	24	28	70
3	Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	2	9	12
4	Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	0	2	3	4	5	12

- Sekarang, yang tampak di samping adalah sekumpulan baris yang benar sebagai header dan semua baris yang tidak dibutuhkan telah dihilangkan.
- Pandas telah mengubah nama kolom yang mengandung nama “countries” dari NaN menjadi Unnamed:0



Hands-on Data Cleaning: Mengganti Nama Kolom

- Untuk mengganti nama kolom, digunakan metode `rename()` DataFrame yg memungkinkan memberi label pada axis berdasarkan pemetaan (dalam kasus ini yaitu dict)
- Mulai dengan mendefinisikan suatu kamus yang memetakan nama kolom saat ini sebagai kunci ke yang lebih dapat digunakan

```
new_names = {'Unnamed: 0': 'Country',
             '? Summer': 'Summer Olympics',
             '01 !': 'Gold',
             '02 !': 'Silver',
             '03 !': 'Bronze',
             '? Winter': 'Winter Olympics',
             '01 !.1': 'Gold.1',
             '02 !.1': 'Silver.1',
             '03 !.1': 'Bronze.1',
             '? Games': '# Games',
             '01 !.2': 'Gold.2',
             '02 !.2': 'Silver.2',
             '03 !.2': 'Bronze.2'}
```

Hands-on Data Cleaning: Mengganti Nama Kolom

- Kemudian, panggil fungsi `rename()` pada objek dimaksud:

```
olympics_df.rename(columns=new_names, inplace=True)
```

- Atur `inplace` menjadi `True`, dengan hasil sbb:

```
olympics_df.head()
```

	Country	Summer Olympics	Gold	Silver	Bronze	Total	Winter Olympics	Gold.1	Silver.1	Bronze.1	Total.1	# Games	Gold.2	Silver.2	Bronze.2	Combined total
0	Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	0	2	2
1	Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	2	8	15
2	Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	24	28	70
3	Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	2	9	12
4	Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	0	2	3	4	5	12

Referensi

- Krensky P. Data Pre Tools: Goals, Benefits, and The Advantage of Hadoop. Aberdeen Group Report. July 2015
- SAS. Data Preparation Challenges Facing Every Enterprise. ebook. December 2017
- <https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/?sh=6e9aa0e36f63>
- <https://improvado.io/blog/what-is-data-preparation>
- <https://searchenterpriseai.techtarget.com/feature/Data-preparation-for-machine-learning-still-requires-humans?>
- <https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e>
- CRISP-DM



Tools Lab Online

- jupyter notebook
- scikit-learn
- pandas
- numpy

Ringkasan

- Data preparation memiliki sebutan lain, diantaranya data pre-processing, data cleaning, data manipulation
- Data preparation mengambil porsi kerja terbanyak dalam data science 60-80%
- Data preparation membutuhkan ketelitian dan kesabaran/kerajinan dari peneliti DS, terutama pemula
- Data Validation merupakan tahapan kritis dari DS namun sering diabaikan para peneliti
- Seleksi Fitur harus dilakukan di awal tahapan data preparation setelah melakukan penentuan metode/teknik sampling
- Data cleaning merupakan pekerjaan yang sangat memerlukan keahlian teknik DS terkait penggunaan tools dan coding
- Kebersihan data merupakan syarat mutlak untuk Model Prediksi yang Baik.



Quiz / Tugas

Quiz dapat diakses melalui <https://spadadikti.id/>



Terima kasih