



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-5

Data Understanding 1: Mengumpulkan Data, Menelaah Data dengan Metode Statistik



[ditjen.dikti](#)



[@ditjendikti](#)



[ditjen.dikti](#)



Ditjen Diktiristek



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



Profil Pengajar: Nama Lengkap dan Gelar Akademik



Poto
Pengajar

Contak Pengajar:

Ponsel:

xxxxxx

Email:

xxxxxxx

Jabatan Akademik:

Latar Belakang Pendidikan:

- S1:
- S2:
- S3:

Riwayat/Pengalaman Pekerjaan:

- Dosen
- XXXX
- XXXX
- XXXX
- XXXX



Unit Kompetensi

- Data Understanding: Mengumpulkan Data
(UK J.62DMloo.004.1 - Mengumpulkan Data)
 - Menentukan kebutuhan data
 - Mengambil data
 - Mengintegrasikan data
- Data Understanding: Menelaah Data dengan Metoda Statistik
(UK J.62DMloo.005.1 - Menelaah Data)
 - Menganalisis tipe dan relasi data
 - Menganalisis karakteristik data
 - Kriteria Unjuk Kerja (KUK) 2.1 Statistika Dasar



Mengumpulkan Data

KODE UNIT : J.62DMI00.004.1

JUDUL UNIT : Mengumpulkan Data

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

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Menentukan kebutuhan data	1.1 Kebutuhan data diidentifikasi sesuai tujuan teknis <i>data science</i> . 1.2 Kebutuhan data diperiksa ketersediaannya sesuai aturan yang berlaku . 1.3 Kebutuhan data ditentukan volumenya sesuai tujuan teknis <i>data science</i> .
2. Mengambil data	2.1 Metode dan tools pengambilan data diidentifikasi sesuai tujuan teknis <i>data science</i> 2.2 <i>Tools</i> pengambilan data ditentukan sesuai tujuan teknis <i>data science</i> 2.3 <i>Tools</i> pengambilan data disiapkan sesuai tujuan teknis <i>data science</i> 2.4 Proses pengambilan data dijalankan sesuai dengan <i>tools</i> yang telah disiapkan
3. Mengintegrasikan data	3.1 Integritas data diperiksa sesuai tujuan teknis <i>data science</i> 3.2 Data diintegrasikan sesuai tujuan teknis <i>data science</i>

Bussiness understanding

1. Konteks variabel
 - 1.1 Kebutuhan data termasuk didalamnya entitas dan atribut data
 - 1.2 Aturan yang berlaku termasuk di dalamnya prosedur dan otorisasi mengakses data. Selain itu, perlu diperhatikan juga aturan penggunaan dari masing-masing situs yang akan diambil datanya.
 - 1.3 Pengambilan data adalah cara mengumpulkan data mentah, termasuk di dalamnya label data yang sesuai tujuan teknis *data science*.
 - 1.4 Metode pengambilan data adalah cara pengambilan data yang berupa otomasi maupun manual (contoh: *survey*, *scraping*, entri data, akses data pihak ketiga, serta tidak terbatas contoh-contoh yang dimaksud).
 - 1.5 *Tools* pengambilan data adalah *tools* yang berupa bahasa pemrograman tertentu, *tools* dari kode sumber terbuka, *tools* dengan lisensi hak milik lainnya (contoh: *scrapy*, *wget*, serta tidak terbatas contoh-contoh yang dimaksud).
 - 1.6 Integritas data adalah kondisi cacat atau tidaknya data akibat proses pengambilan atau pemindahan data.



Menelaah Data

KODE UNIT : J.62DMI00.005.1

JUDUL UNIT : Menelaah Data

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

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Menganalisis tipe dan relasi data	1.1 Tipe data yang terkumpul diidentifikasi sesuai tujuan teknis 1.2 Nilai atribut data yang terkumpul diuraikan sesuai dengan batasan konteks bisnisnya 1.3 Relasi antar data yang terkumpul diidentifikasi sesuai dengan tujuan teknis
2. Menganalisis karakteristik data	2.1 Karakteristik data yang terkumpul disajikan dengan deskripsi statistik dasar 2.2 Karakteristik data yang terkumpul disajikan dengan visualisasi grafik 2.3 Hasil penyajian data dianalisis karakteristiknya untuk telaah data
3. Membuat laporan telaah data	3.1 <u>Hasil analisis didokumentasikan dalam bentuk laporan sesuai dengan tujuan teknis</u> 3.2 Hipotesis disusun berdasar hasil analisis sesuai tujuan teknis <i>data science</i>

1. Konteks variabel

- 1.1 Data yang terkumpul adalah data yang sudah diintegrasikan dari proses mengumpulkan data pada tahap sebelumnya yang sesuai kebutuhan *data science*.
- 1.2 Tipe data termasuk di dalamnya tipe dan nilai datanya.
- 1.3 Deskripsi statistik dasar adalah analisis statistik meliputi nilai maksimum, minimum, rerata, median, modus, *skewness*, persentil, distribusi, *outliers* dan lain sejenisnya.

Data Understanding Documentation



Bahan Bacaan

- Modul Pembelajaran Data Understanding
- Joel Grus, "Data Science from Scratch: First Principles with Python", 2nd Edition, O'Reilly 2019.
- Charu C. Aggarwal, "Data Mining: The Textbook", Springer, 2015.
- Matt Taddy, "Business Data Science", McGraw-Hill, 2019.



Deskripsi modul dan tujuan pembelajaran

- Modul ini berisi penjelasan mengenai konsep dan teknik pengambilan dan telaah data (*data gathering and understanding*). Teknik-teknik yang dibahas dibatasi pada yang bersifat non visual menggunakan statistika. Teknik-teknik visualisasi dijelaskan secara terpisah di modul 07.
- Setelah menyelesaikan modul ini, peserta diharapkan mampu:
 - melakukan pengambilan data untuk proses sains data dari sumber data terbuka, baik secara manual maupun secara programatik menggunakan library Pandas;
 - melakukan telaah data dengan beberapa metode statistika



Outline

- Apa itu telaah data (data understanding)?
- Sumber, susunan, tipe, dan model data
- Pengambilan data
- Telaah data dasar

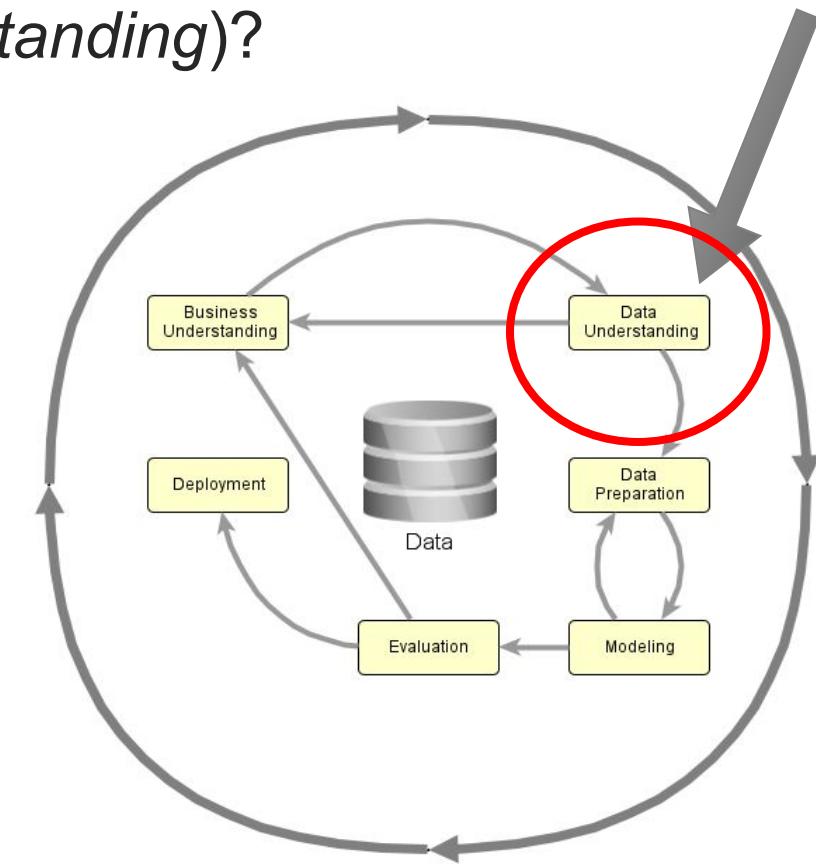


Apa itu Telaah Data (*Data Understanding*)?



Apa itu telaah data (*data understanding*)?

- Dilakukan setelah problem bisnis telah didefinisikan sebagai hasil tahapan business understanding.
- Tujuan: mendapatkan gambaran utuh atas data.
- Dilanjutkan ke persiapan data (data preparation), jika pemahaman awal data cukup atau kembali ke business understanding jika definisi permasalahan bisnis harus direvisi.





Mengapa perlu data understanding?

- Data = bahan mentah solusi AI
- Data dari masing-masing sumber belum tentu dapat langsung dipakai karena:
 - maksud dan tujuan data berbeda-beda
 - keadaan asal terpisah-pisah atau justru terintegrasi secara ketat.
 - tingkat kekayaan (*richness*) berbeda-beda
 - tingkat keandalan (*reliability*) berbeda-beda
- Data understanding memberikan gambaran awal tentang:
 - kekuatan data
 - kekurangan dan batasan penggunaan data
 - tingkat kesesuaian data dengan masalah bisnis yang akan dipecahkan
 - ketersediaan data (terbuka/tertutup, biaya akses, dsb.)



Bagian-bagian proses telaah data

Identifikasi "titik sentuh" data dengan proses bisnis

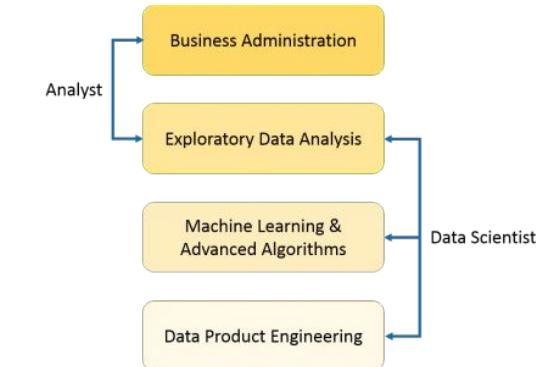
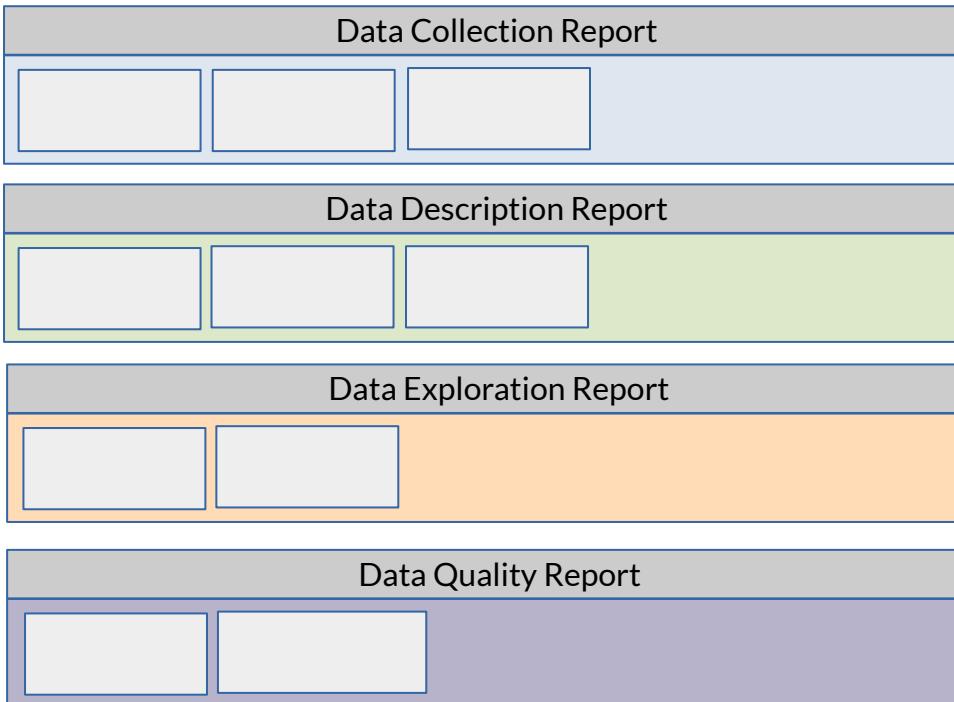
Penentuan sumber utama data dan cara aksesnya

Asesmen nilai tambah bisnis dari data

Identifikasi sumber data tambahan untuk perbaikan



Data Understanding



PHASE	CRISP-DM PLANNING														
	WEEK 1				WEEK 2				WEEK 3						
	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
Business Understanding															
Data Understanding															
Data Preparation															
Modeling															
Evaluation															



Data Understanding Documentation

• Data Collection Report

- Data Sources
- Existing data
- Purchased Data
- Additional Data
- Attribute of data
- Enough data?
- Merging various data

• Data Description Report

- Describing Data
- Amount of Data
- Value Types
- Coding Scheme
- Data Quantity (Format, method to capture, how large)
- Data Quality (characteristic, type of data)
- Compute basic statis of key attributes
- Priority of attributes

• Data Exploration Report

- Hypothesis of data
- Promising attributes for further analysis
- Exploration of new characteristics of data
- How change initial hypothesis
- Identify subset of data for later use
- Compare with goals

• Data Quality Report

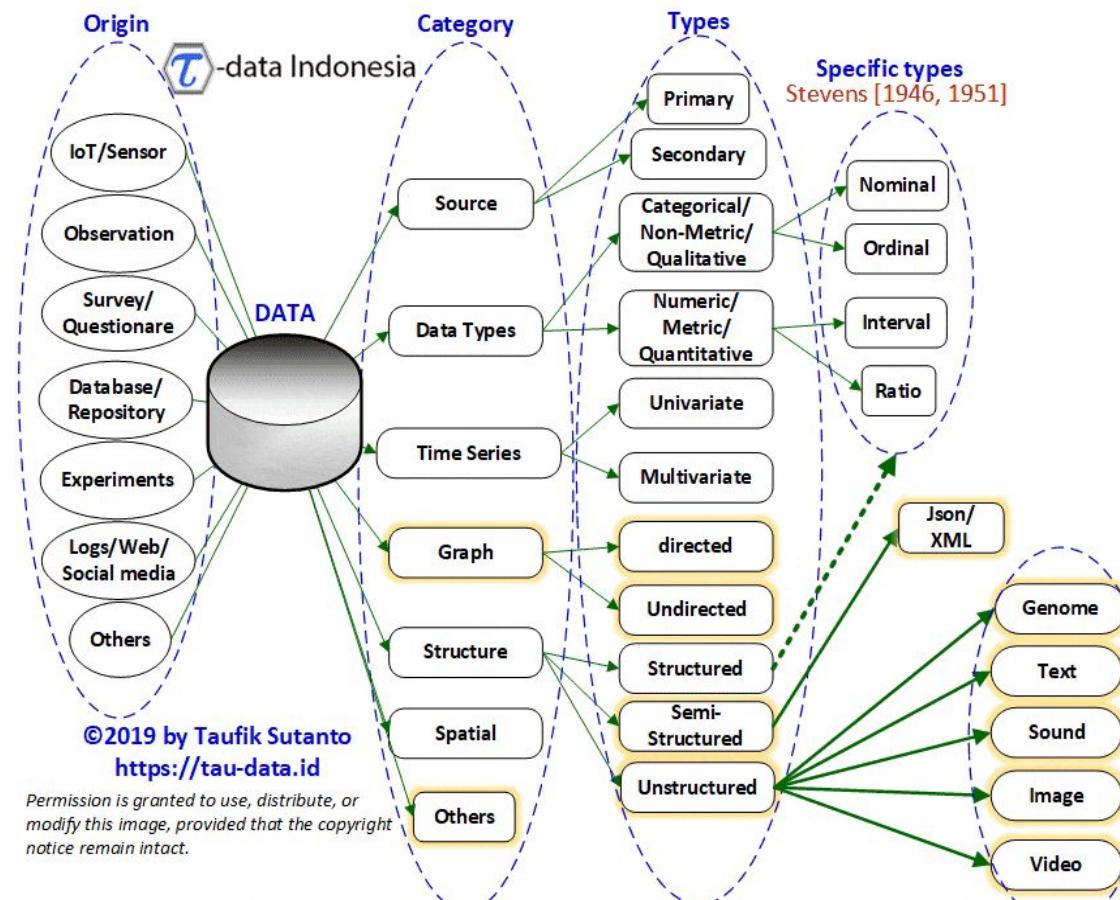
- Missing Data
- Data Errors
- Measurement Errors
- Missing attributes abd blank fields
- Spelling inconsistencies?
- Noise
- Plausibility check for values (conflict of values)
- Considering to exclude data that has no impact to hypothesis
- Are data stored in flat files? Are delimeter consistent, How is each record



Sumber, Susunan, Tipe dan Model Data



Peta Tipe Data





Sumber data

Internal sources Spreadsheets (Excel, CSV, JSON, etc.)
Databases: can be queried via SQL, etc.

Text documents

Multimedia documents (audio, video)

External sources Open data repositories

Public domain web pages



Sumber data daring

- Portal Satu Data Indonesia (<https://data.go.id>)
- Portal Data Jakarta (<https://data.jakarta.go.id>)
- Portal Data Bandung (<http://data.bandung.go.id>)
- Badan Pusat Statistik (<https://www.bps.go.id>)
- Badan Informasi Geospasial (<https://tanahair.indonesia.go.id/>)
- UCI Machine Learning repository (<https://archive.ics.uci.edu/ml/index.php>)
- Kaggle (<https://www.kaggle.com/datasets>)
- World Bank Open Data (<https://data.worldbank.org>)
- UNICEF Data (<https://data.unicef.org>)
- WHO Open Data (<https://www.who.int/data>)
- IBM Data Asset eXchange (<https://developer.ibm.com/exchanges/data/>)
- DBPedia (<https://www.dbpedia.org/resources/>)
- Wikidata (<https://www.wikidata.org/>)



Sumber data daring

- Cari via Google Dataset Search: <https://datasetsearch.research.google.com>

The screenshot shows the Google Dataset Search interface. At the top left is the Google logo. To its right are icons for help, a menu, and sign-in. Below the header is a large title "Dataset Search". A search bar contains the placeholder "Search for Datasets" and a magnifying glass icon. Below the search bar is a text field with the placeholder "Try coronavirus covid-19 or education outcomes site:data.gov.". At the bottom of the search area is a link "Learn more about Dataset Search." and a globe icon.

Susunan data

Butir data (datum): satuan terkecil data;
satu nilai untuk satu variable tertentu

Data: kumpulan butir data yang
membawa satu kesatuan makna
(mendeskripsikan satu objek) tertentu.

Himpunan data (dataset): kumpulan
data.

Metadata: data yang menjelaskan data
yang lain.

symboling	normalized-losses	make	fuel-type
3	?	alfa-romero	gas
3	?	alfa-romero	gas
1	?	alfa-romero	gas
2	164	audi	gas
2	164	audi	gas

"make":

- tipe: string,
- deskripsi: nama pabrikan merek kendaraan

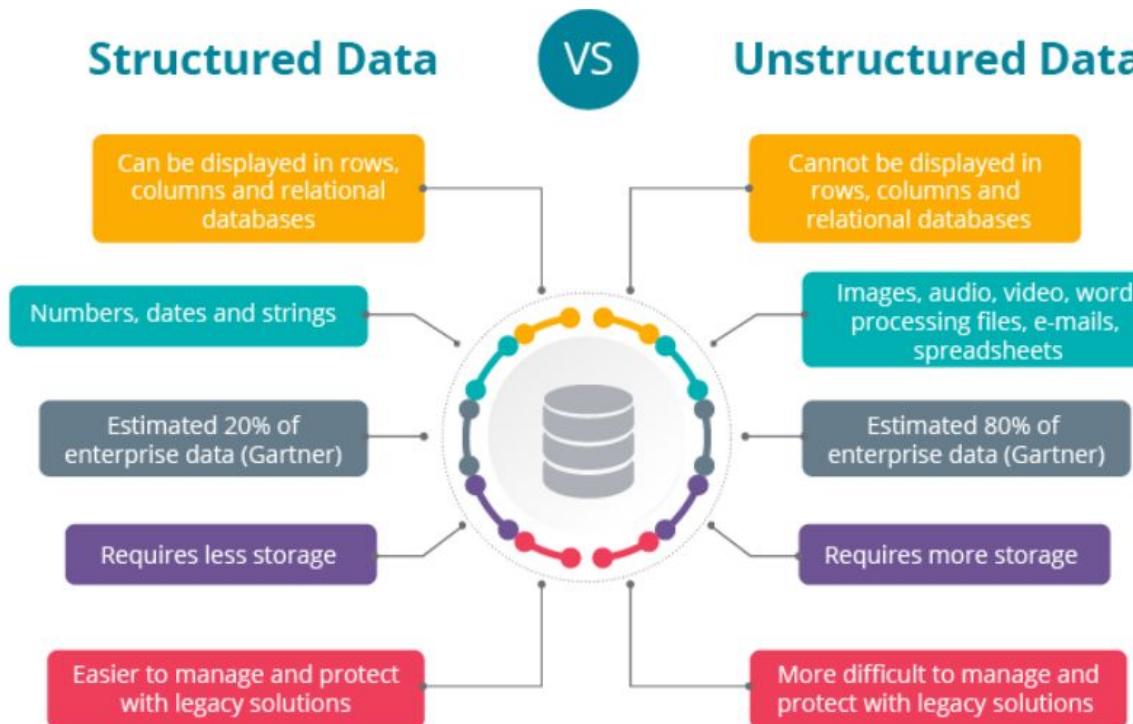


Tipe data berdasarkan susunannya

	Data terstruktur (structured data)	Data tak terstruktur (unstructured data)
Sifat	<ul style="list-style-type: none">Model data terdefinisi sebelumnyaFormat butir data (biasanya) teks.Antar butir data dibedakan dengan jelas.Ekstraksi/kueri langsung cukup mudah.	<ul style="list-style-type: none">Model data tidak terdefinisi sebelumnyaFormat butir data (biasanya) teks, citra, suara, video, dan format lainnya.Antar butir data tidak cukup jelas terbedakan karena ketidakteraturan dan ambiguitas.Ekstraksi/kueri langsung cukup sulit.
Contoh	Data tabular, data berorientasi objek, <i>time series</i>	Data teks dalam dokumen teks bebas, data audio, data video.

Data semi-terstruktur (*semi-structured data*): Data terstruktur yang tidak mengikuti model struktur tabular yang seperti pada basis data relasional, namun tetap mengandung *tags* atau penanda lainnya yang dapat memisahkan elemen-elemen semantik pada data serta mengatur hierarki antara butir-butir datanya.

Structured vs Unstructured Data (modul 8/9)



Sumber:
<https://www.knowledgehut.com/blog/data-science/role-of-unstructured-data-in-data-science>



Contoh Unstructured Data^(modul 8/9)

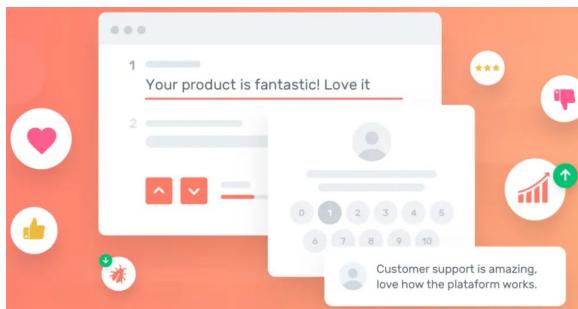
email

The screenshot shows an email inbox with three messages:

- The New York Times: News - Breaking News: E.U. leaders agreed to a \$857 billion stimulus package that includes unprecedented ...
- SEMrush Team: Business - On Page SEO Checker: 33 New Optimization Ideas!
- Zoom: Business - Your meeting attendees are waiting!

email: semi-structured data; isi email: unstructured data

review pelanggan



web pages

The screenshot shows a web page titled "monkeylearn.com/unstructured-data/". It features a header with a search bar, navigation links (Definition, Types, Importance, Analysis), and a "Try It Out Now" button. The main content area discusses "customer feedback analysis" and "Webpages". It explains that the vast internet creates unstructured data at breakneck speed, mentioning text, images, audio, video, and all manner of content. It also notes that while the structure of web pages is written in HTML code, this doesn't actually explain the content of the pages. A note at the bottom states that it can be useful to mine, extract, and organize this data to find information.

media sosial

The screenshot shows a search results page for "#Covid_19usa" on a platform like Twitter. The search bar contains "#Covid_19usa". Below it, a search result for "Coronavirus @COVID_19usa" is shown, featuring a profile picture of a virus cell and the handle @COVID_19usa.

sumber:
<https://monkeylearn.com/unstructured-data/>



Tipe data berdasarkan Sifatnya

- **Data dikotomi**, merupakan data yang bersifat pilah satu sama lain, misalnya suku, agama, jenis kelamin, pendidikan, dan lain sebagainya.
- **Data diskrit**, merupakan data yang proses pengumpulan datanya dijalankan dengan cara menghitung atau membilang. Seperti, jumlah anak, jumlah penduduk, jumlah kematian dan sebagainya.
- **Data kontinum**, merupakan data pengumpulan datanya didapatkan dengan cara mengukur dengan alat ukur yang memakai skala tertentu. Seperti misalnya, Suhu, berat, bakat, kecerdasan, dan lainnya.



Tipe data berdasarkan Cara Pengumpulan

- **Data primer**, merupakan data yang didapatkan dari sumber pertama, atau dapat dikatakan pengumpulannya dilakukan sendiri oleh si peneliti secara langsung, seperti hasil wawancara dan hasil pengisian kuesioner (angket).
- **Data sekunder**, merupakan data yang didapatkan dari sumber kedua. Menurut Purwanto (2007), data sekunder yaitu data yang dikumpulkan oleh orang atau lembaga lain. Data sekunder adalah data yang digunakan atau diterbitkan oleh organisasi yang bukan pengolahnya (Soeratno dan Arsyad (2003;76).



Tipe butir data (1)

	Nominal/kategorikal	Ordinal	Interval	Rasio
Sifat himpunan asal	Diskret, tidak terurut	Diskret, terurut	Kontinu/numerik, terurut, perbedaan menunjukkan selisih	Kontinu/numerik, terurut, nilai menunjukkan rasio terhadap kuantitas satuan/unit di jenis yang sama
Contoh	Warna (merah, hijau, biru)	Nilai huruf mahasiswa (A, B, C, D, E)	Suhu dalam Celcius, tanggal dalam kalender tertentu	Panjang jalan, suhu dalam Kelvin
Ukuran data menyatakan ...	Membership	Membership, comparison	Membership, comparison, difference	Membership, comparison, difference, magnitude
Operasi matematika	=, ≠	=, ≠, <, >	=, ≠, <, >, +, -, ×, ÷	=, ≠, <, >, +, -, ×, ÷



Tipe butir data (2)

	Nominal/kategorikal	Ordinal	Interval	Rasio
Representasi nilai tipikal	Modus	Modus, median	Modus, median, rerata aritmetik	Modus, median, rerata aritmatik, rerata geometris, rerata harmonis
Representasi sebaran	Grouping	Grouping, rentang (<i>range</i>), rentang antar kuartil	Grouping, rentang (<i>range</i>), rentang antar kuartil, varians, simpangan baku	Grouping, rentang (<i>range</i>), rentang antar kuartil, varians, simpangan baku, koefisien variasi
Memiliki nol sejati yang menyatakan nilai mutlak terbawah.	Tidak	Tidak	Tidak	Ya



Tipe data berdasarkan Waktunya

- Data Cross Section

Data cross-section adalah data yang menunjukkan titik waktu tertentu.

Contohnya laporan keuangan per 31 Desember 2020, data pelanggan PT.

Data Indah bulan mei 2004, dan lain sebagainya.

- Data Time Series / Berkala

Data berkala adalah data yang datanya menggambarkan sesuatu dari waktu ke waktu atau periode secara historis. Contoh data time series adalah data perkembangan nilai tukar dollar amerika terhadap rupiah tahun 2016 - 2020



Contoh model data: Tabular

- Terdiri dari N buah rekord (*record*)
- Masing-masing record mengandung D buah atribut
- Record = baris, *data point*, instans, *example*, transaksi, tupel, entitas, objek, vector fitur.
- Atribut = kolom, *field*, dimensi, fitur.
- Atribut yang sama untuk setiap record biasanya diasumsikan memiliki tipe butir data yang sama.
- Struktur dapat bersifat ketat/strict (contoh: basis data relasional) atau longgar/loose (contoh: Excel *spreadsheet*).
- Tergantung keketatan strukturnya, bisa ada bahasa kueri formal untuk mengakses butir-butir data di dalamnya (contoh: SQL).

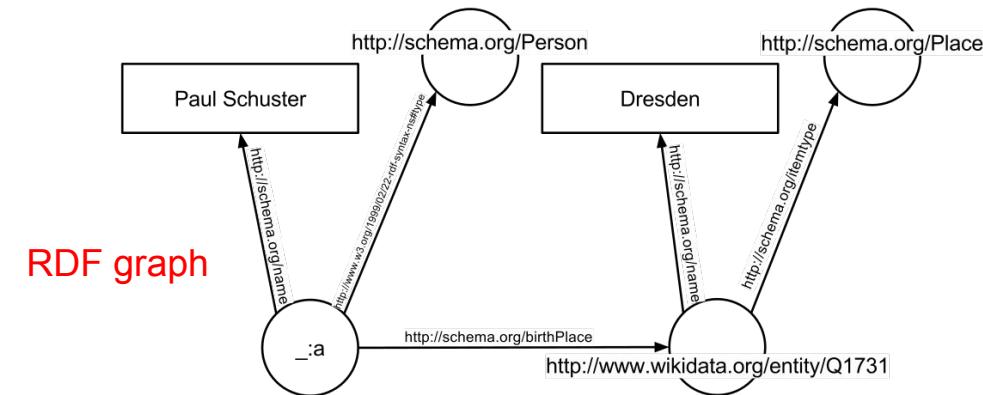
symboling	normalized-losses	make
3 ?		alfa-romero
3 ?		alfa-romero
1 ?		alfa-romero
2		164 audi
2		164 audi

Contoh model data: Graf/Jejaring

- Tersusun dari simpul-simpul (*nodes*) dan sisi/koneksi antar simpul (*edges*)
- Satu node (biasanya) mewakili satu record
- Dapat mengekspresikan relasi antar record secara eksplisit.
- Termasuk model data graf adalah model data hierarkii/pohon, model data berorientasi objek (*object-oriented data model*).
- Model data graf modern:
 - *Property graph*
 - *Resource description framework (RDF)*



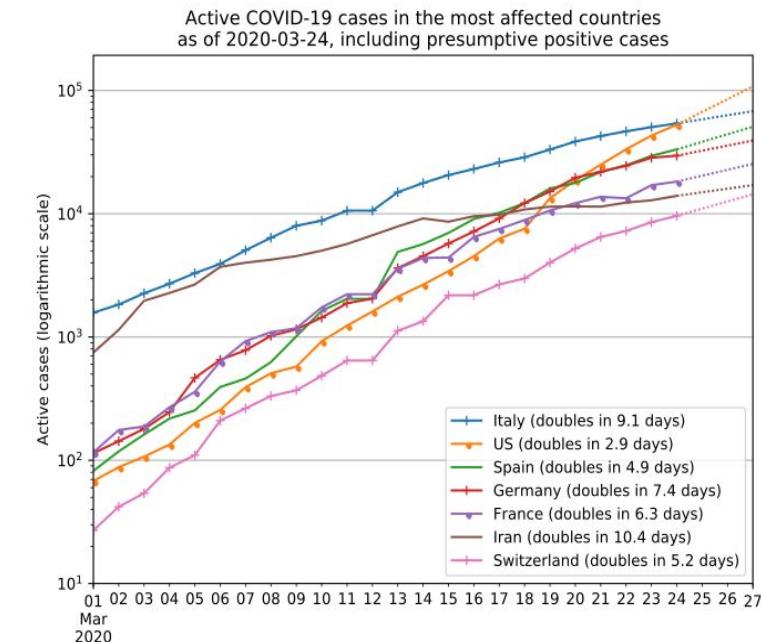
Property graph



RDF graph

Contoh model data: Sekuens

- Tersusun dari rekord-rekord yang terhubung secara sekuensial.
- Contoh: data dari sensor suhu selama suatu rentang waktu.
- Struktur tersirat dari urutan kemunculan rekord
- Rekaman audio dan video dapat dipandang sebagai data sekuens, namun setiap recordnya sendiri bersifat tidak terstruktur.
- Atribut kontekstual mendefinisikan basis dependensi tersirat. (Contoh: time stamp pada sensor suhu)
- Atribut behavioral: butir-butir data yang nilainya diperoleh dalam suatu konteks tertentu (Contoh: besarnya suhu).
- Jika atribut kontekstualnya adalah waktu/time stamp, maka data sekuens disebut *time series*.





Pengambilan Data





Pengambilan Data

- Pengambilan data secara manual.
- Pengambilan data melalui API
 - Contoh melalui API Kaggle
 - Contoh melalui API Portal Data Bandung
- Pengambilan data melalui *web scraping*
- Pengambilan data melalui akses langsung ke basis data relasional yang ada.



Pengambilan data secara manual

Cari data di sumber data



Unduh/salin data ke *local machine*



Muat (*load*) data ke pengolah data
→ Jupyter Notebook



Mengambil data (secara manual) dari Kaggle

- Kita akan mengakses data dari "Goal Dataset – Top 5 European Leagues" dari Kaggle.
- Kunjungi Kaggle.com dan login (buat akun jika perlu)
- Lakukan pencarian "goal dataset top 5 European leagues"
- Klik "Goal Dataset – Top 5 European Leagues"

The screenshot shows the Kaggle search interface with the query "goal dataset top 5 european leagues". The results page displays several datasets, with one specific entry highlighted by a red rectangle:

- Football Data: Expected Goals and Other Metrics** by Sergi Lehkyi (a year ago, 1 MB, 93 views)
- The Beautiful Game - Analysis of Football Events** by Ahmed Youssef (3 years ago, 2m to run, R, 102 views)
- Goal Dataset - Top 5 European Leagues** by shreyansh khandelwal (a month ago, 174 KB, 6 views)
- Football Events** by Alin Secareanu (a month ago, 200 views)

On the left, there are filters for Date (Last 90 days), Viewed By You (Viewed, Not Viewed), Dataset Size (small, medium), Dataset File Types (csv, xlsx, sqlite), and Dataset License (Other, Commercial, Non-Commercial). A "More" link is also present under Dataset File Types.



Data Explorer

383.68 KB

- ☰ Bundesliga-goalScorer(20-21).csv
- ☰ LaLiga-goalScorer(20-21).csv
- ☰ Ligue_1-goalScorer(20-21).c...
- ☰ Serie_A-goalScorer(20-21)....
- ☰ epl-goalScorer(20-21).csv

< epl-goalScorer(20-21).csv (73.58 KB)

Detail Compact Column

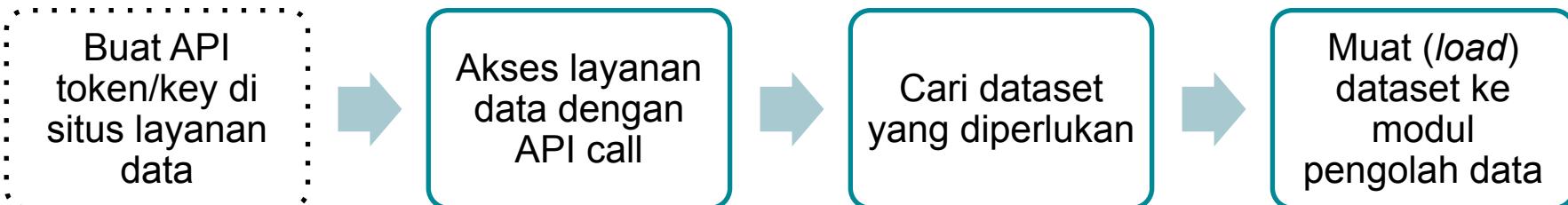
10 of 19 columns ▾

#		# id	A player_name	# games	# time	#
0		521				
1		647	Harry Kane	35	3097	23
2		1250	Mohamed Salah	37	3085	22
3		1228	Bruno Fernandes	37	3117	18
4		453	Son Heung-Min	37	3139	15
5		822	Patrick Bamford	38	3085	17

- Di halaman data explorer, pilih "epl-goalScorer (20-21).csv"
- Unduh data dengan mengklik tombol unduh di bagian kanan dan simpan di folder kerja Anda.

Pengambilan data melalui API

- Data dapat diambil melalui *application programming interface* (API).
 - API disediakan oleh beberapa layanan data seperti Kaggle.
 - API token/key (mungkin) diperlukan untuk mengakses data via API.
 - Proses pembuatan API token/key (jika perlu) dirinci di dokumentasi masing-masing layanan.





Mengambil data dengan API dari Kaggle (1)

- Nyalakan Jupyter Notebook di folder kerja Anda, lalu buka atau buat satu skrip baru (Python 3).
- Instal kaggle library (mis: dengan pip)

```
In [1]: !pip install kaggle
```



Mengambil data dengan API dari Kaggle (2)

- Login ke Kaggle, klik foto profil Anda (di kanan atas), kemudian klik 'Your Profile' untuk membuka halaman profil Anda.
- Pada halaman profil Anda, klik tab 'Account'. Geser ke bawah sedikit, dan Anda akan menemukan tombol 'Create New API Token'

The screenshot shows the 'Account' tab selected in the navigation bar. Below it, there are sections for 'Phone Verification' (status: Not verified), 'Email Preferences' (instructions to control via notification settings), and 'API'. The 'API' section contains instructions about Kaggle's beta API and links to documentation. A prominent red circle highlights the 'Create New API Token' button, which is located below these instructions. Other buttons in the row include 'Edit Public Profile' (in a black button) and 'Expire API Token'.



Mengambil data dengan API dari Kaggle (3)

- Klik 'Create New API Token'. Jika tombol tidak berfungsi, klik 'Expire API Token' lebih dahulu.
 - Browser akan mengunduh file `kaggle.json` ke folder unduhan (Downloads) Anda.
- Kaggle API secara default mengasumsikan bahwa file `kaggle.json` tersebut berada di dalam folder:
`~/ .kaggle/` (Linux/Mac) atau
`C:\Users\<Windows-username>\.kaggle\` (Windows)
 - Jika folder tersebut belum ada, buat dulu dengan perintah `mkdir` di shell/command line.
 - Pindahkan file `kaggle.json` ke folder tersebut (menggunakan File/Windows Explorer atau melalui perintah `mv` atau `move` di shell)



Mengambil data dengan API dari Kaggle (4)

- Kaggle API memiliki empat perintah
 - kaggle competitions {list, files, download, submit, submissions, leaderboard}
 - kaggle datasets {list, files, download, create, version, init}
 - kaggle kernels {list, init, push, pull, output, status}
 - kaggle config {view, set, unset}
- Dokumentasi Kaggle API dapat dilihat di
<https://github.com/Kaggle/kaggle-api>
- Untuk keperluan modul ini, kita hanya menggunakan perintah kaggle datasets



Mengambil data dengan API dari Kaggle (5)

- Untuk melakukan pencarian dataset: kaggle datasets list -s <keyword>
 - Jika terjadi masalah gagal akses, dsb., bisa dicoba dengan membuat ulang API Token.
- Nama dataset berada di kolom ref pada tabel output pencarian. Misalnya kita ingin mengunduh "Goal Dataset – Top 5 European Leagues, maka nama dataset adalah shreyanshkhanelwal/goal-dataset-top-5-european-leagues .

In [2]: !kaggle datasets list -s "goal leagues"

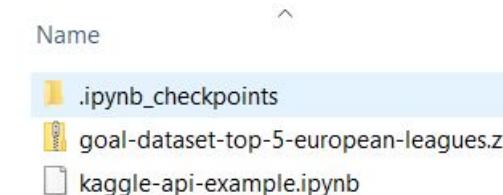
ref ated	downloadCount	voteCount	usabilityRating	title	size	lastUpd
slehkyi/extended-football-stats-for-european-leagues-xg -02 17:28:39	2733	94	1.0	Football Data: Expected Goals and Other Metrics	1MB	2020-08
secareanualin/football-events -25 01:19:19	19416	525	0.7647059	Football Events	21MB	2017-01
shreyanshkhanelwal/goal-dataset-top-5-european-leagues -23 21:20:09	25	6	0.5294118	Goal Dataset - Top 5 European Leagues	174KB	2021-05
chaibapat/fantasy-premier-league -16 18:56:26	1466	31	0.85294116	Fantasy Premier League - 2016/2017	476MB	2017-05
yamaerenay/most-popular-soccer-leagues -01 16:59:30	78	5	1.0	Most Popular Soccer Leagues	30KB	2020-08



Mengambil data dengan API dari Kaggle (6)

- Unduh dataset yang diinginkan dengan perintah `kaggle datasets download`

```
In [3]: !kaggle datasets download shreyanshkhanelwal/goal-dataset-top-5-european-leagues
```



- Dataset akan terunduh di folder aktif dalam bentuk file terkompresi zip.

- Selanjutnya, kita ekstraksi dataset tersebut dengan perintah `unzip`, dan dataset berupa berkas-berkas csv siap digunakan.

- Berkas csv dapat langsung dimuat ke Pandas DataFrame

```
In [4]: !unzip goal-dataset-top-5-european-leagues.zip
```

```
Archive: goal-dataset-top-5-european-leagues.zip
inflating: Bundesliga-goalScorer(20-21).csv
inflating: LaLiga-goalScorer(20-21).csv
inflating: Ligue_1-goalScorer(20-21).csv
inflating: Serie_A-goalScorer(20-21).csv
inflating: epl-goalScorer(20-21).csv
```



Mengambil data dari Portal Satu Data Bandung dengan API (1)

- Portal Satu Data Bandung (<http://data.bandung.go.id>) juga merupakan sumber data terbuka yang dapat diakses melalui API berbasis CKAN. Dokumentasi umum CKAN dapat diakses di <https://docs.ckan.org>.

The screenshot shows the homepage of the OpenData Kota Bandung portal. At the top, there is a navigation bar with links for Dataset, Organisasi, Grup, Tentang, Bantuan, and Versi 2.0. Below the navigation is a search bar labeled "Cari Data Apa ?" with a magnifying glass icon. The main header features the "OPENDATA KOTA BANDUNG" logo and the "PEMERINTAH KOTA BANDUNG" seal. To the right of the seal is the "Diskominfo KOTA BANDUNG" logo. The page has two main sections: "Portal Data Kota Bandung" which describes the portal as "Data Kota Bandung dalam 1 Portal" and "Portal Data Kota Bandung statistics" which displays four key numbers: 3k datasets, 74 organizations, 8 groups, and 11 visualisasi.

CKAN: open data portal



Mengambil data dari Portal Satu Data Bandung dengan API (2)

- Kita gunakan Python library `requests`, dan `json`
- Daftar 3014 dataset diperoleh via API call ke http://data.bandung.go.id/api/3/action/package_list

```
In [2]: ds_list_response = requests.get("http://data.bandung.go.id/api/3/action/package_list")
ds_list_result = json.loads(ds_list_response.text)
len(ds_list_result['result']), ds_list_result
```

```
Out[2]: (3014,
{'help': 'http://data.bandung.go.id/api/3/action/help_show?name=package_list',
 'success': True,
 'result': ['10-besar-penyakit-rawat-inap-tahun-2015-bandung',
 '10-besar-penyakit-rawat-jalan-rskia-kota-bandung',
 '10-kasus-penyakit-tertinggi-di-rsud-kota-bandung-berdasarkan-jenis-kelamin-tahun-2016',
 '10-kasus-penyakit-tertinggi-di-rsud-kota-bandung-berdasarkan-jenis-kelamin-tahun-2017',
 '10-kasus-penyakit-tertinggi-di-rsud-kota-bandung-berdasarkan-jenis-kelamin-tahun-2018',
 '10-kasus-penyakit-tertinggi-di-rsud-kota-bandung-berdasarkan-jenis-kelamin-tahun-2019',
 '10-kasus-penyakit-tertinggi-di-rsud-kota-bandung-berdasarkan-kelompok-usia-tahun-2016',
 '10-kasus-penyakit-tertinggi-di-rsud-kota-bandung-berdasarkan-kelompok-usia-tahun-2017',
 '10-kasus-penyakit-tertinggi-di-rsud-kota-bandung-berdasarkan-kelompok-usia-tahun-2018',
 '10-kasus-penyakit-tertinggi-di-rsud-kota-bandung-berdasarkan-kelompok-usia-tahun-2019',
 '20-penyakit-terbesar-di-puskesmas-kota-bandung',
 'akreditasi-sekolah-dasar-negeri-berdasarkan-kecamatan-di-kota-bandung',
 'akreditasi-sekolah-dasar-swasta-berdasarkan-kecamatan-di-kota-bandung',
 'akreditasi-sekolah-menengah-pertama-negeri-berdasarkan-kecamatan-di-kota-bandung',
 'akreditasi-sekolah-menengah-pertama-swasta-berdasarkan-kecamatan-di-kota-bandung',
 'alamat-kantor-kecamatan-di-kota-bandung',
 'alamat-kantor-kelurahan-di-kota-bandung']}
```

`requests.get`
mengirim HTTP Get
request ke layanan

Hasil dalam format
JSON di-load
sebagai Python
dictionary dengan
`json.loads`

Daftar nama-nama
dataset ada pada
key 'result'



Mengambil data dari Portal Satu Bandung dengan API (3)

Selain mem-filter langsung pada daftar nama-nama dataset di slide sebelumnya, kita dapat melakukan search dengan memanggil API http://data.bandung.go.id/api/3/action/package_search dengan parameter 'q' untuk frasa yang dicari. Contoh untuk mencari "sekolah dasar":

```
In [3]: search_response = requests.get("http://data.bandung.go.id/api/3/action/package_search",
                                   params=[('q', 'sekolah dasar')])
search_result = json.loads(search_response.text)
print('Result count:', search_result['result']['count'])
print(len(search_result['result']['results']))
search_result
```

```
Result count: 47
10
```

```
Out[3]: {'help': 'http://data.bandung.go.id/api/3/action/help_show?name=package_search',
          'success': True,
          'result': {'count': 47,
                     'sort': 'score desc, metadata_modified desc',
                     'facets': {},
                     'results': [{"license_title": 'Creative Commons Attribution',
                                 'maintainer': 'Open Data Kota Bandung',
                                 'relationships_as_object': [],
                                 'private': False,
                                 'maintainer_email': 'data@bandung.go.id',
                                 'num_tags': 5,
                                 'id': 'cb838eb9-5e60-44e0-bbe8-9275b77b2fe9',
                                 'metadata_created': '2019-09-29T05:16:48.171920',
                                 'metadata_modified': '2020-08-10T02:39:37.600396',
                                 'author': 'Open Data Kota Bandung',
                                 'author_email': 'data@bandung.go.id',
                                 'state': 'active'}]}
```

Namun, keterbatasan API membuat detil hanya dikembalikan pada 10 hasil pencarian, walaupun seharusnya ada 47.



Mengambil data dari Portal Satu Bandung dengan API (4)

Di sini, kita filter nama dataset secara manual, misalnya yang mengandung "**sekolah**" dan "**dasar**"

```
In [5]: sd_datasets = [ds_name for ds_name in ds_list_result['result'] if 'sekolah' in ds_name and 'dasar' in ds_name]
sd_datasets
```

```
Out[5]: ['akreditasi-sekolah-dasar-negeri-berdasarkan-kecamatan-di-kota-bandung',
 'akreditasi-sekolah-dasar-swasta-berdasarkan-kecamatan-di-kota-bandung',
 'akreditasi-sekolah-menengah-pertama-negeri-berdasarkan-kecamatan-di-kota-bandung',
 'akreditasi-sekolah-menengah-pertama-swasta-berdasarkan-kecamatan-di-kota-bandung',
 'jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah',
 'jumlah-guru-sd-swasta-di-kota-bandung-berdasarkan-sekolah',
 'jumlah-guru-smp-swasta-di-kota-bandung-berdasarkan-sekolah',
 'jumlah-pelayanan-kesehatan-gigi-dan-mulut-di-sekolah-dasar',
 'kemiskinan-kota-bandung-berdasarkan-indikator-angka-partisipasi-sekolah',
 'rekapitulasi-jumlah-pendaftar-smp-negeri-pilihan-pertama-berdasarkan-jarak-rumah-ke-sekolah',
 'rekapitulasi-jumlah-pendaftar-smp-negeri-sebagai-pilihan-kedua-berdasarkan-jarak-rumah-ke-sekolah',
 'rekapitulasi-pendaftar-smp-negeri-sebagai-pilihan-kedua-berdasarkan-asal-sekolah',
 'rekapitulasi-pendaftar-smp-negeri-sebagai-pilihan-pertama-berdasarkan-asal-sekolah',
 'sekolah-dasar-di-kecamatan-andir',
 'sekolah-dasar-di-kecamatan-antapani',
 'sekolah-dasar-di-kecamatan-arcamanik',
 'sekolah-dasar-di-kecamatan-astanaanyar',
 'sekolah-dasar-di-kecamatan-babakan-ciparay',
 'sekolah-dasar-di-kecamatan-bandung-kidul',
 'sekolah-dasar-di-kecamatan-bandung-kulon',
 'sekolah-dasar-di-kecamatan-bandung-wetan',
 'sekolah-dasar-di-kecamatan-batununggal',
 'sekolah-dasar-di-kecamatan-batununggal-kota-bandung',
 'sekolah-dasar-di-kecamatan-bojongloa-kaler',
 'sekolah-dasar-di-kecamatan-bojongloa-kidul']
```



Mengambil data dari Portal Satu Bandung dengan API (5)

Misal kita menginginkan dataset "jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah". Kita pakai API call http://data.bandung.go.id/api/3/action/package_show dengan parameter nama dataset.

```
In [5]: ds_name = 'jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah'
ds_response = requests.get("http://data.bandung.go.id/api/3/action/package_show",
                           params=[('id',ds_name)])
ds_info = json.loads(ds_response.text)
len(ds_info['result']['resources']), ds_info
```

```
Out[5]: (1,
         {'help': 'http://data.bandung.go.id/api/3/action/help_show?name=package_show',
          'success': True,
          'result': {'license_title': 'Creative Commons Attribution',
                     'maintainer': 'Open Data Kota Bandung',
                     'relationships_as_object': [],
                     'private': False,
                     'maintainer_email': 'data@bandung.go.id',
                     'num_tags': 4,
                     'id': '5c9b9612-0fa7-4a93-88bd-12022ca39217',
                     'metadata_created': '2018-11-29T07:53:38.493370',
                     'metadata_modified': '2019-10-11T07:49:12.108563',
                     'author': 'Open Data Kota Bandung',
                     'author_email': 'data@bandung.go.id',
                     'state': 'active',
                     'version': '',
                     'creator_user_id': '0dcc4b7c-b933-4390-b726-fde75b0dc5ae',
                     'type': 'dataset',
                     'resources': [{cache_last_updated': None,
                                   'package_id': '5c9b9612-0fa7-4a93-88bd-12022ca39217',
                                   'webstore_last_updated': None,
                                   'datastore_active': False,
                                   'id': '85550991-9192-4059-84d0-2e1f38546cdb',
                                   'size': None,
                                   'state': 'active',
                                   'hash': ''}]}), ds_info
```

Setiap dataset dapat terdiri dari satu atau beberapa berkas (misal: csv, xls). Info ini disimpan pada key 'resources' yang berupa Python list.

Kebetulan dataset ini hanya memiliki 1 berkas saja.



Mengambil data dari Portal Satu Bandung dengan API (6)

Kita akses daftar berkas untuk dataset "jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah".

```
In [6]: ds_urls = [d['url'] for d in ds_info['result']['resources']]  
ds_urls
```

```
Out[6]: ['http://data.bandung.go.id/dataset/5c9b9612-0fa7-4a93-88bd-12022ca39217/resource/85550991-9192-4059-84d0-2e1f38546cdb/downloa  
d/jumlah-guru-di-sd-negeri-kota-bandung-2018.csv']
```

Kita unduh berkas-berkas dengan URL di atas.

Library `tqdm` dipakai untuk menampilkan kemajuan unduhan.

Setelah selesai, data tersimpan di folder tempat notebook dijalankan.

```
In [7]: from tqdm import tqdm  
for url in ds_urls:  
    fname = url[url.rfind('/'):]  
    with open(fname, "wb") as handle:  
        resp = requests.get(url, stream=True)  
        for data in tqdm(resp.iter_content()):  
            handle.write(data)  
        print(fname, 'saved.')
```

```
14314it [00:00, 89701.71it/s]
```

```
jumlah-guru-di-sd-negeri-kota-bandung-2018.csv saved.
```



Pengambilan data melalui API

- Kaggle dan beberapa layanan data lainnya menyediakan akses melalui API.
- Langkah-langkah mengakses API biasanya melalui proses pembuatan API token/API key yang dirinci di dokumentasi masing-masing layanan.
- Selain API, teknik pengambilan data yang bersifat lanjut mencakup *web scraping* serta akses data langsung dari basis data relasional.



Mengambil data dengan *Web Scraping*

- *Web scraping* = mengekstraksi data secara langsung dari suatu halaman web.
- Langkah-langkah umum (contoh detil dapat dilihat di <https://realpython.com/beautiful-soup-web-scraping-python/>)
 - Tentukan URL halaman web (HTML) yang akan di-*scrape*.
 - Gunakan fungsi `requests.get` untuk mengakses URL tersebut. Teks HTML akan tersimpan pada atribut `text` dari object yang dikembalikan `requests.get`.
 - Lakukan *parsing* pada HTML dengan library `beautifulsoup` untuk memperoleh tabel data yang diinginkan (dengan mengekstraksi elemen-elemen HTML yang relevan).



Mengambil data dari basis data relasional (RDB)

- Data juga dapat bersumber dari basis data relasional (RDB) organisasi.
- Langkah-langkah umum:
 - Import pandas
 - Import library penghubung RDB, misal: `mysql.connector` untuk MySQL
 - Gunakan method `connect` dari penghubung RDB untuk membuka koneksi ke RDB.
 - Siapkan SQL query dalam string.
 - Gunakan `pandas.read_sql` dengan argument string SQL query dan koneksi RDB untuk mengeksekusi SQL dan memuat hasilnya ke dalam `DataFrame`.
 - Tutup koneksi.
- Proses antara membuka hingga menutup koneksi biasanya ditaruh dalam blok `try-except`
- Pembukaan koneksi membutuhkan kredensial (username, password) ke RDB yang di-*hardcode* secara langsung. Ini dapat disembunyikan dengan teknik pengamanan yang tidak dibahas di sini.
- Contoh singkat dapat dilihat di:
<https://medium.com/analytics-vidhya/importing-data-from-a-mysql-database-into-pandas-data-frame-a06e392d27d7>



Memuat Data ke Pandas





Memuat data ke Pandas (1)

- Nyalakan Jupyter Notebook di folder kerja Anda.
- Buka atau buat baru suatu skrip ipynb (Python 3)
- Import pandas dan numpy. (Pastikan sudah terinstal sebelumnya).
- Load file CSV yang sudah diunduh sebelumnya (pada contoh "Mengambil Data secara Manual") ke dalam sebuah DataFrame
 - Gunakan perintah `read_csv(...)`

```
1 import pandas as pd
```

```
1 dataset = 'cloth_data.csv'
```

```
1 df = pd.read_csv(dataset)
```



Memuat data ke Pandas (2)

- Method head() dan tail() pada DataFrame membantu kita menampilkan 5 beberapa baris pertama/terakhir dari data yang kita muat.

```
df.head(3)
```

Unnamed: 0	id	player_name	games	time	goals	xG	assists
0	0	647	Harry Kane	35	3097	23	22.174859
1	1	1250	Mohamed Salah	37	3085	22	20.250847
2	2	1228	Bruno Fernandes	37	3117	18	16.019454

```
df.head()
```

Unnamed: 0	id	player_name	games	time	goals	xG	assists
0	0	647	Harry Kane	35	3097	23	22.174859
1	1	1250	Mohamed Salah	37	3085	22	20.250847
2	2	1228	Bruno Fernandes	37	3117	18	16.019454
3	3	453	Son Heung-Min	37	3139	17	11.023287
4	4	822	Patrick Bamford	38	3085	17	18.401863



Telaah Data





Mengungkap tipe-tipe data dari setiap kolom

- Atribut `dtypes` pada DataFrame berisi tipe data dari setiap kolom.
- Lihat Pandas User Guide untuk detail setiap tipe.
- `dtype : object` di akhir output `dtypes` mewakili Series yang merupakan objek Python yang dikembalikan oleh `dtypes` itu sendiri (bukan bagian dari tipe kolom manapun).
- Dalam contoh ini, terlihat bahwa 2 kolom pertama hanyalah ID numerik yang biasanya tidak memiliki makna riil. Maka, kita ambil bagian DataFrame mulai dari kolom "player_name" (untuk *zero-based index*, kita pakai kolom ke-2 dst).

print(df.dtypes)				
df_noid = df.iloc[:, 2:]				
	player_name	games	time	goals
0	Harry Kane	35	3097	23 22.17
1	Mohamed Salah	37	3085	22 20.28
2	Bruno Fernandes	37	3117	18 16.03
3	Son Heung-Min	37	3139	17 11.02
4	Patrick Bamford	38	3085	17 18.40
...
517	Jaden Philogene-Bidace	1	1	0 0.00
518	Gaetano Berardi	2	113	0 0.07
519	Anthony Elanga	1	67	0 0.00
520	Femi Seriki	1	1	0 0.00



Deskripsi statistik data

DataFrame method `describe()` menampilkan statistik dasar setiap kolom data yang bertipe numerik, mencakup banyaknya data (**count**), rerata aritmetik (**mean**), simpangan baku (**std**), nilai terkecil (**min**), kuartil pertama (**25%**), kuartil kedua/median (**50%**), kuartil ketiga (**75%**), dan nilai terbesar (**max**).

```
df_noid.describe()
```

	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	npg
count	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.
mean	19.643678	1420.068966	1.862069	2.000806	1.289272	1.376029	17.379310	12.963602	2.061303	0.091954	1.668582
std	11.619836	1031.604819	3.338851	3.317946	2.083350	1.886510	21.572664	16.164361	2.203661	0.295800	2.909929
min	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	10.000000	470.250000	0.000000	0.074668	0.000000	0.049245	2.000000	1.000000	0.000000	0.000000	0.000000
50%	21.000000	1342.000000	1.000000	0.737295	0.000000	0.691122	10.000000	7.000000	2.000000	0.000000	0.500000
75%	30.000000	2319.000000	2.000000	2.053378	2.000000	2.050509	23.750000	19.000000	3.000000	0.000000	2.000000
max	38.000000	3420.000000	23.000000	22.174859	14.000000	11.474996	138.000000	95.000000	12.000000	2.000000	19.000000



Konsep: Rerata Aritmetik

- Nilai rerata yang lazim dipahami kebanyakan orang.
- Rerata aritmetik dari sekumpulan bilangan = jumlah semua bilangan tersebut dibagi dengan banyaknya bilangan dalam kumpulan.
- Diberikan sekumpulan N buah bilangan $S = \{x_1, \dots, x_N\}$, rerata aritmetik μ_S atau \bar{x} dari S didefinisikan sebagai:

$$\mu_S = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i = \frac{x_1 + \dots + x_N}{N}$$

- Merupakan salah satu ukuran pusat data (tendensi sentral) yang dapat dipakai untuk data bertipe interval dan rasio.
- **Sifat:** total jarak setiap bilangan x_i terhadap rerata aritmetik \bar{x} adalah 0.
- Dapat dipakai sebagai bilangan yang mewakili keseluruhan kumpulan, sepanjang distribusi datanya **tidak** bersifat *skew* (asimetris).



Konsep: Simpangan Baku

- Simpangan baku (*standard deviation*) adalah salah satu ukuran sebaran data.
- Dipakai untuk data bertipe interval dan rasio.
- Untuk kumpulan bilangan $S = \{x_1, \dots, x_N\}$ dengan rerata aritmetik μ_S , simpangan baku σ_S dari S adalah

$$\sigma_S = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu_S)^2} = \sqrt{\frac{(x_1 - \mu_S)^2 + \dots + (x_N - \mu_S)^2}{N}}$$

- Kuadrat dari σ_S , yakni σ_S^2 disebut sebagai **varians**
- Nilai simpangan baku
 - besar = data secara umum tersebar jauh dari nilai rerata aritmetik
 - kecil = data secara umum terkumpul dekat dengan nilai rerata aritmetik
- Simpangan baku dapat pula dipandang sebagai derajat ketidakpastian pengukuran data
 - Contoh: pada pengukuran berulang dengan suatu instrument yang sama, jika simpangan baku data hasil pengukuran bernilai besar, berarti presisi pengukuran rendah.



Konsep: Median dan Kuartil

- Kuartil pertama (Q_1): nilai data sehingga 25% dari keseluruhan data bernilai lebih kecil darinya.
- Kuartil kedua (Q_2) atau median: nilai data sehingga separuh dari data yang ada bernilai lebih kecil darinya.
 - Dapat dipakai sebagai ukuran pusat data (tendensi sentral) sebagai alternatif dari rerata (khususnya jika distribusi data bersifat *skewed*).
- Kuartil ketiga (Q_3): nilai data sehingga 75% dari keseluruhan data bernilai lebih kecil darinya.
- Kuartil dapat dipakai untuk data bertipe ordinal, interval, dan rasio.



Deskripsi statistik data

Gunakan `describe(include='all')` jika ingin menampilkan juga statistik kolom yang bertipe non-numerik, mencakup juga berapa banyak nilai unik dalam kolom (**unique**), nilai modus (**top**), serta frekuensi modus (**freq**).

df_noid.describe(include='all')										ards	position	team_title	npg	npxG	xGChain	xGBuildup
	player_name	games	time	goals	xG	assists	xA	st	sh	1000	522	522	522.000000	522.000000	522.000000	522.000000
count	522	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	NaN	14	28	NaN	NaN	NaN	NaN
unique	522	Nan	Nan	Nan	Nan	Nan	Nan	Nan	Nan	NaN	M S	Everton	Nan	Nan	NaN	NaN
top	Joel Ward	Nan	Nan	Nan	Nan	Nan	Nan	Nan	Nan	NaN	106	28	Nan	Nan	NaN	NaN
freq	1	Nan	Nan	Nan	Nan	Nan	Nan	Nan	Nan	954	NaN	NaN	1.668582	1.821450	5.663368	3.455060
mean	NaN	19.643678	1420.068966	1.862069	2.000806	1.289272	1.376029	17.379	1800	NaN	NaN	NaN	2.909929	2.931176	5.600249	3.376584
std	NaN	11.619836	1031.604819	3.338851	3.317946	2.083350	1.886510	21.572	1000	NaN	NaN	NaN	0.000000	0.000000	0.000000	0.000000
min	NaN	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000	1000	NaN	NaN	NaN	0.000000	0.074668	1.191391	0.720353
25%	NaN	10.000000	470.250000	0.000000	0.074668	0.000000	0.049245	2.000	1000	NaN	NaN	NaN	0.500000	0.715585	4.252738	2.656397
50%	NaN	21.000000	1342.000000	1.000000	0.737295	0.000000	0.691122	10.000	1000	NaN	NaN	NaN	2.000000	1.945799	8.308002	5.254647
75%	NaN	30.000000	2319.000000	2.000000	2.053378	2.000000	2.050509	23.750	1000	NaN	NaN	NaN	19.000000	19.130183	28.968234	18.323006
max	NaN	38.000000	3420.000000	23.000000	22.174859	14.000000	11.474996	138.000	1000	NaN	NaN	NaN	19.000000	19.130183	28.968234	18.323006



Konsep: Modus

- Modus (*mode*): nilai yang paling sering muncul pada sekumpulan data.
- Dipakai sebagai ukuran pusat data (tendensi sentral) untuk data bertipe nominal/kategoris.
 - Tidak dijamin unik dalam suatu distribusi data (bisa ada lebih dari satu modus dalam suatu distribusi).
 - Merupakan nilai yang berpeluang paling tinggi didapatkan ketika data di-*sample*.
- Contoh:
 - Himpunan data {1,2,2,3,4,4,7,8} memiliki dua modus: 2 dan 4.
- Jika data mengikuti distribusi kontinu, misal
 - {0.935, ..., 1.134, ..., 2.643, ..., 3.459, ..., 3.995, ...}maka secara statistik, tidak boleh diasumsikan akan ada dua data yang bernilai persis sama.
 - Definisi modus standar menjadi tidak bermakna.
 - Pendekatan 1: lakukan diskretisasi (dibahas di modul Data Preparation), sehingga didapat data bertipe nominal, lalu dicari modusnya.
 - Pendekatan 2: gunakan teknik *kernel density estimation* (tidak dibahas di sini).



Fungsi statistik dalam Pandas

count	Number of non-NA observations
sum	Sum of values
mean	Mean of values
mad	Mean absolute deviation
median	Arithmetic median of values
min	Minimum
max	Maximum
mode	Mode
abs	Absolute Value
prod	Product of values
quantile	Sample quantile (value at %), 1st quartile = quantile(0.25)

std	Bessel-corrected sample standard deviation
var	Unbiased variance
sem	Standard error of the mean
skew	Sample skewness (3rd moment)
kurt	Sample kurtosis (4th moment)
cumsum	Cumulative sum
cumprod	Cumulative product
cummax	Cumulative maximum
cummin	Cumulative minimum



Contoh fungsi statistik setiap kolom (yang *applicable*)

df_noid.mean()

```
games           19.643678
time            1420.068966
goals            1.862069
xG              2.000806
assists          1.289272
xA              1.376029
shots            17.379310
key_passes       12.963602
yellow_cards     2.061303
red_cards         0.091954
npg              1.668582
npxG             1.821450
xGChain          5.663368
xGBuildup        3.455060
dtype: float64
```

df_noid.sum()

```
player_name      Harry Kane Mohamed Salah Bruno Fernandes Son Heun...
games                  10254
time                  741276
goals                  972
xG                     1044.420572
assists                 673
xA                     718.287269
shots                  9072
key_passes                6767
yellow_cards               1076
red_cards                  48
npg
npxG
xGChain
xGBuildup
dtype: object
position          FF M SM SF M SF SF SF SF M SF SF SF SF ...
team_title        Tottenham Liverpool Manchester United Tottenham Le...
npg
npxG
xGChain
xGBuildup
dtype: object
950.7971
2956.278233
1803.541131
```



Contoh fungsi statistik setiap kolom (yang *applicable*)

```
df_noid.median()
```

```
games           21.000000
time          1342.000000
goals          1.000000
xG             0.737295
assists        0.000000
xA             0.691122
shots          10.000000
key_passes     7.000000
yellow_cards   2.000000
red_cards      0.000000
npg            0.500000
npxG           0.715585
xGChain        4.252738
xGBuildup     2.656397
dtype: float64
```

```
df_noid.std()
```

```
games           11.619836
time          1031.604819
goals          3.338851
xG             3.317946
assists        2.083350
xA             1.886510
shots          21.572664
key_passes     16.164361
yellow_cards   2.203661
red_cards      0.295800
npg            2.909929
npxG           2.931176
xGChain        5.600249
xGBuildup     3.376584
dtype: float64
```

```
df_noid.quantile(0.75) # 3rd quartile
```

```
games           30.000000
time          2319.000000
goals          2.000000
xG             2.053378
assists        2.000000
xA             2.050509
shots          23.750000
key_passes     19.000000
yellow_cards   3.000000
red_cards      0.000000
npg            2.000000
npxG           1.945799
xGChain        8.308002
xGBuildup     5.254647
Name: 0.75, dtype: float64
```

Menentukan pencilan (secara kasar) berdasarkan statistik

- **3-sigma rule:** Jika data kira-kira terdistribusi normal:
 - x_i adalah pencilan jika $x_i < \mu_S - 2\sigma_S$ atau $x_i > \mu_s + 2\sigma_S$
→ peluang bahwa data berjarak ke rerata lebih jauh dari 2 kali simpangan baku adalah 4.55%.
 - x_i adalah pencilan jika $x_i < \mu_S - 3\sigma_S$ atau $x_i > \mu_s + 3\sigma_S$
→ peluang bahwa data berjarak ke rerata lebih jauh dari 3 kali simpangan baku adalah 0.27%.
 - Kekurangan: (i) asumsi distribusi normal (belum tentu!), (ii) rerata dan simpangan baku dipengaruhi nilai pencilan itu sendiri, dan (iii) tidak dapat mendeteksi pencilan jika jumlah data sedikit (*small sample size*).
- **Tukey's fences:** memakai **rentang antarkuartil** (*interquartile range*) $IQR = Q_3 - Q_1$.
 - x_i adalah pencilan jika $x_i < Q_1 - 1.5(IQR)$ atau $x_i > Q_3 + 1.5(IQR)$.
 - x_i adalah pencilan ekstrim jika $x_i < Q_1 - 3(IQR)$ atau $x_i > Q_3 + 3(IQR)$.
- Metode-metode lain (mungkin lebih baik): Visualisasi, Grubb's test, Dixon's Q test, Algoritma Expectation Maximization, Jarak k-Nearest Neighbor, *local outlier factor* berbasis *density* (variasi *density-based clustering*), dll.



Mencari pencilan dengan *Tukey's fences* (1)

- Hitung IQR tiap kolom
- Variabel `q1` dan `q3` adalah Pandas Series berisi nilai-nilai kuartil pertama dan kuartil ketiga dari kolom-kolom numerik data.
- Variabel `iqr` adalah Pandas Series berisi nilai rentang antar kuartil untuk kolom-kolom numerik data.

```
q1 = df_noid.quantile(0.25)
q3 = df_noid.quantile(0.75)
iqr = q3 - q1
iqr
```

```
games           20.000000
time          1848.750000
goals           2.000000
xG            1.978711
assists        2.000000
xA             2.001264
shots          21.750000
key_passes     18.000000
yellow_cards   3.000000
red_cards      0.000000
npg            2.000000
npxG           1.871131
xGChain        7.116612
xGBuildup     4.534294
dtype: float64
```



Mencari pencilan dengan Tukey's fences (2)

- Filter nilai-nilai di `df_noid` yang termasuk pencilan sesuai kriteria Tukey.
- Sebelum filter dihitung, harus dilakukan `join` dulu antara DataFrame `df_noid` dan Series `iqr`

```
df_noid_align, iqr_new = df_noid.align(iqr, axis=1, copy=False, join='outer')
outlier_filter = (df_noid1 < q1 - 1.5 * iqr_new) | (df_noid1 > q3 + 1.5 * iqr_new)
outlier_filter
```

	assists	games	goals	key_passes	npg	npxG	player_name	position	red_cards	shots	team_title	time	xA	xG	xGBuildup	xGChain	yellow_c
0	True	False	True	True	True	True		False	False	False	True	False	True	True	False	True	F
1	False	False	True	True	True	True		False	False	False	True	False	False	True	True	True	F
2	True	False	True	True	True	True		False	False	False	True	False	False	True	True	True	F
3	True	False	True	True	True	True		False	False	False	True	False	False	True	True	True	F
4	True	False	True	False	True	True		False	False	False	True	False	False	False	True	True	F
...	
517	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	F
518	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	F
519	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	F
520	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	F
521	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	F

522 rows × 17 columns





Mencari pencilan dengan *Tukey's fences* (3)

- Contoh: mencari nama pemain dengan jumlah *assist* yang termasuk pencilan.
 - Assist* = umpan yang menghasilkan gol.

```
df_noid1[outlier_filter['assists']] \\\n    .loc[:, ['player_name', 'assists']] \\n    .sort_values(by=['assists'], ascending=False)
```

	player_name	assists
0	Harry Kane	14
2	Bruno Fernandes	12
58	Kevin De Bruyne	11
3	Son Heung-Min	10
51	Jack Grealish	10
6	Jamie Vardy	9
15	Marcus Rashford	9
57	Raphinha	9
41	Jack Harrison	8
281	Aaron Cresswell	8
83	Pascal Groß	8
49	Timo Werner	8
32	James Ward-Prowse	7
26	Roberto Firmino	7
16	Sadio Mané	7
130	Trent Alexander-Arnold	7
204	Andrew Robertson	7
4	Patrick Bamford	7
358	Lucas Digne	7
42	Bertrand Traoré	6



Value_counts

- value_counts() menghasilkan frekuensi setiap nilai unik di dalam kolom.
- Yang tertinggi count-nya adalah merupakan modus pada kolom tersebut.
- Ada data dengan dua/tiga nama tim karena ada pemain yang bermain di dua/tiga klub dalam musim yang sama (ada transfer pemain).

In [18]: df['team_title'].value_counts()

Out[18]: West Bromwich Albion 28

Everton 28

Fulham 27

Wolverhampton Wanderers 27

Southampton 27

Sheffield United 27

Manchester United 27

Liverpool 27

Leicester 27

Brighton 26

Arsenal 26

Newcastle United 26

Chelsea 25

Burnley 25

Tottenham 24

Manchester City 24

Crystal Palace 24

West Ham 23

Leeds 23

Aston Villa 23

West Bromwich Albion,West Ham 1

Everton,Southampton 1

Arsenal,West Bromwich Albion 1

Chelsea,Fulham 1

Aston Villa,Chelsea 1

Arsenal,Newcastle United 1

Liverpool,Southampton 1

Arsenal,Brighton 1



Analisis dengan groupby

- Method `groupby` memungkinkan analisa dilakukan secara per kelompok nilai atribut tertentu.
Misal: rerata dan simpangan baku gol per tim.

In [30]: `df.groupby('team_title')['goals'].std()`

Out[30]: team_title

Arsenal	3.352381
Arsenal,Brighton	NaN
Arsenal,Newcastle United	NaN
Arsenal,West Bromwich Albion	NaN
Aston Villa	3.696489
Aston Villa,Chelsea	NaN
Brighton	2.158703
Burnley	2.475210
Chelsea	2.350177
Chelsea,Fulham	NaN
Crystal Palace	2.901461
Everton	3.467727
Everton,Southampton	NaN
Fulham	1.439175
Leeds	4.153193
Leicester	4.020602
Liverpool	4.931439
Liverpool,Southampton	NaN
Manchester City	3.867132
Manchester United	4.317855
Newcastle United	2.483174

In [29]: `df.groupby('team_title')['goals'].mean()`

Out[29]: team_title

Arsenal	1.961538
Arsenal,Brighton	0.000000
Arsenal,Newcastle United	8.000000
Arsenal,West Bromwich Albion	0.000000
Aston Villa	2.130435
Aston Villa,Chelsea	3.000000
Brighton	1.500000
Burnley	1.280000
Chelsea	2.240000
Chelsea,Fulham	1.000000
Crystal Palace	1.625000
Everton	1.607143
Everton,Southampton	3.000000
Fulham	0.925926
Leeds	2.608696
Leicester	2.370370
Liverpool	2.370370
Liverpool,Southampton	3.000000
Manchester City	3.208333
Manchester United	2.518519
Newcastle United	1.384615



Korelasi Pearson antara kolom-kolom numerik

- Method corr() menghasilkan tabel korelasi Pearson antar kolom-kolom numerik.
- Rentang nilai: antara -1 dan 1.
- 1 = korelasi negatif, 0 = tidak ada korelasi linear, +1 = korelasi positif.

In [23]: df.loc[:, 'games'].corr()

Out[23]:

	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	npg	npxG	xGChain	xGBt
games	1.000000	0.944591	0.439730	0.463869	0.504168	0.562806	0.599164	0.617867	0.565963	0.160326	0.437110	0.465546	0.726598	0.61
time	0.944591	1.000000	0.398930	0.411203	0.473555	0.516638	0.529534	0.575065	0.592223	0.186333	0.392631	0.408231	0.703801	0.71
goals	0.439730	0.398930	1.000000	0.932798	0.617490	0.607330	0.873363	0.567752	0.097151	0.053679	0.971591	0.905710	0.727953	0.21
xG	0.463869	0.411203	0.932798	1.000000	0.636205	0.627495	0.910214	0.570488	0.093761	0.048815	0.894286	0.979218	0.763909	0.21
assists	0.504168	0.473555	0.617490	0.636205	1.000000	0.885850	0.721220	0.835299	0.209349	-0.021444	0.587316	0.615503	0.752587	0.41
xA	0.562806	0.516638	0.607330	0.627495	0.885850	1.000000	0.759568	0.946506	0.243912	0.006284	0.585152	0.611100	0.814487	0.51
shots	0.599164	0.529534	0.873363	0.910214	0.721220	0.759568	1.000000	0.743370	0.249957	0.073932	0.852989	0.901386	0.843152	0.41
key_passes	0.617867	0.575065	0.567752	0.570488	0.835299	0.946506	0.743370	1.000000	0.343357	0.022780	0.539726	0.545537	0.807958	0.61
yellow_cards	0.565963	0.592223	0.097151	0.093761	0.209349	0.243912	0.249957	0.343357	1.000000	0.165064	0.093270	0.089065	0.401884	0.51
red_cards	0.160326	0.186333	0.053679	0.048815	-0.021444	0.006284	0.073932	0.022780	0.165064	1.000000	0.055542	0.047354	0.104005	0.11
npg	0.437110	0.392631	0.971591	0.894286	0.587316	0.585152	0.852989	0.539726	0.093270	0.055542	1.000000	0.913496	0.720978	0.21
npxG	0.465546	0.408231	0.905710	0.979218	0.615503	0.611100	0.901386	0.545537	0.089065	0.047354	0.913496	1.000000	0.763481	0.21
xGChain	0.726598	0.703801	0.727953	0.763909	0.752587	0.814487	0.843152	0.807958	0.401884	0.104005	0.720978	0.763481	1.000000	0.81
xGBuildup	0.697196	0.731377	0.290990	0.282746	0.473254	0.547983	0.448197	0.618754	0.562467	0.167660	0.284135	0.273090	0.802073	1.00

<

>



Quiz / Tugas

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



Terima kasih