DATA UNDERSTANDING AND PREPARATION

Mulia Sulistiyono, M.Kom
muliasulistiyono@amikom.ac.id

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.

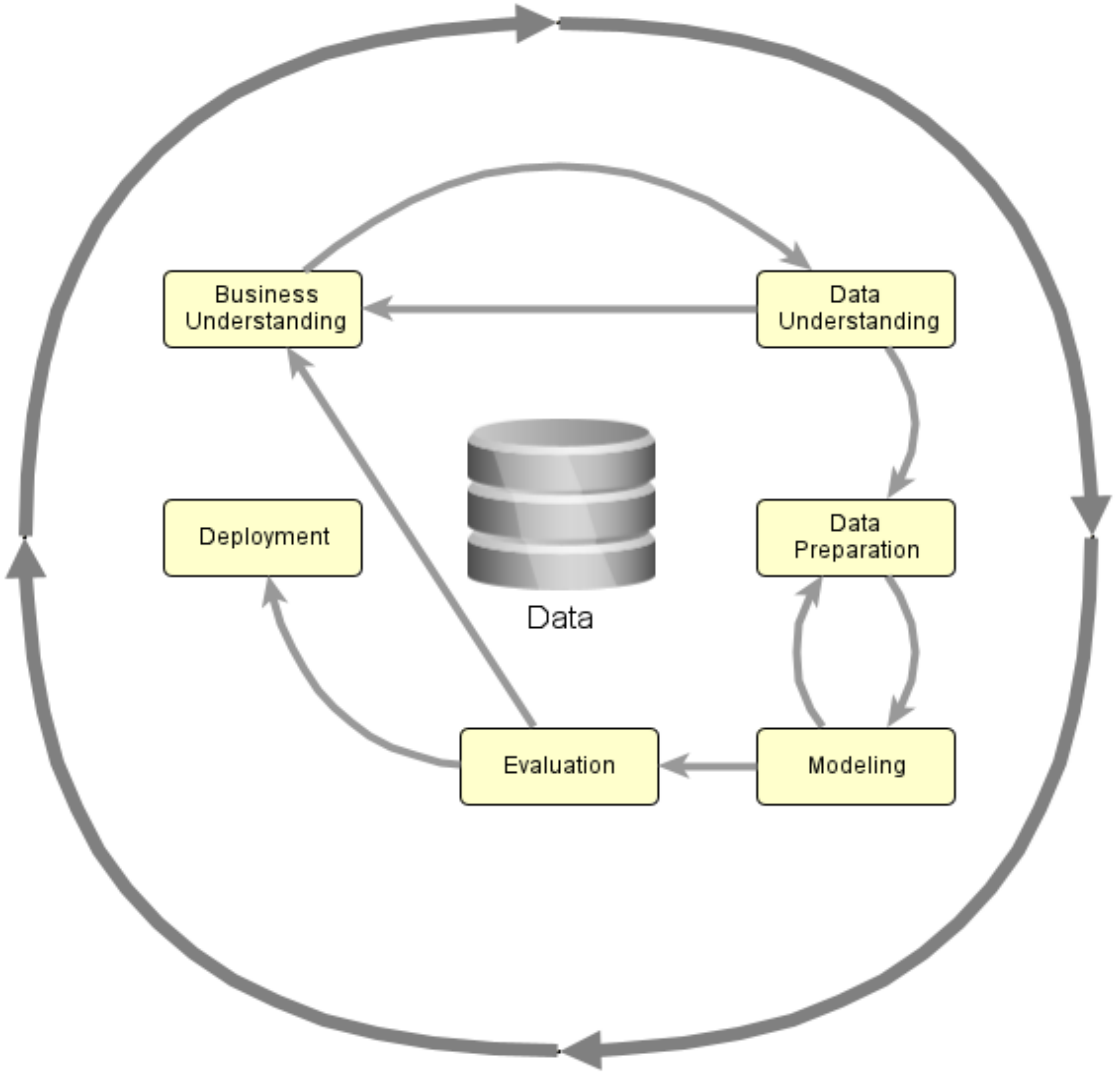
Outline

- 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 terdefinisikan 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

Sumber, Susunan, Tipe dan Model Data

Sumber data

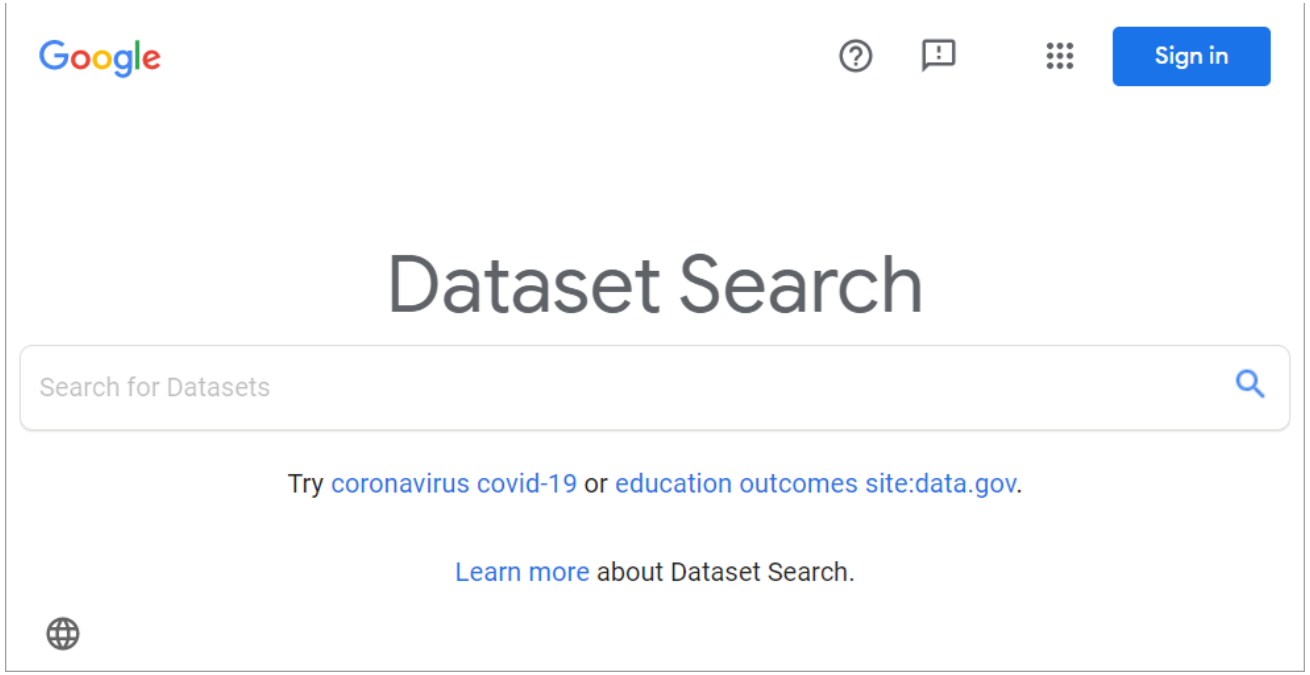
Internal sources	Spreadsheets (Excel, CSV, JSON, etc.)
	Databases: can be queried via SQL, etc.
	Text documents
	Multimedia documents (audio, video)
External sources	Open data repositories
	Public domain web pages

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



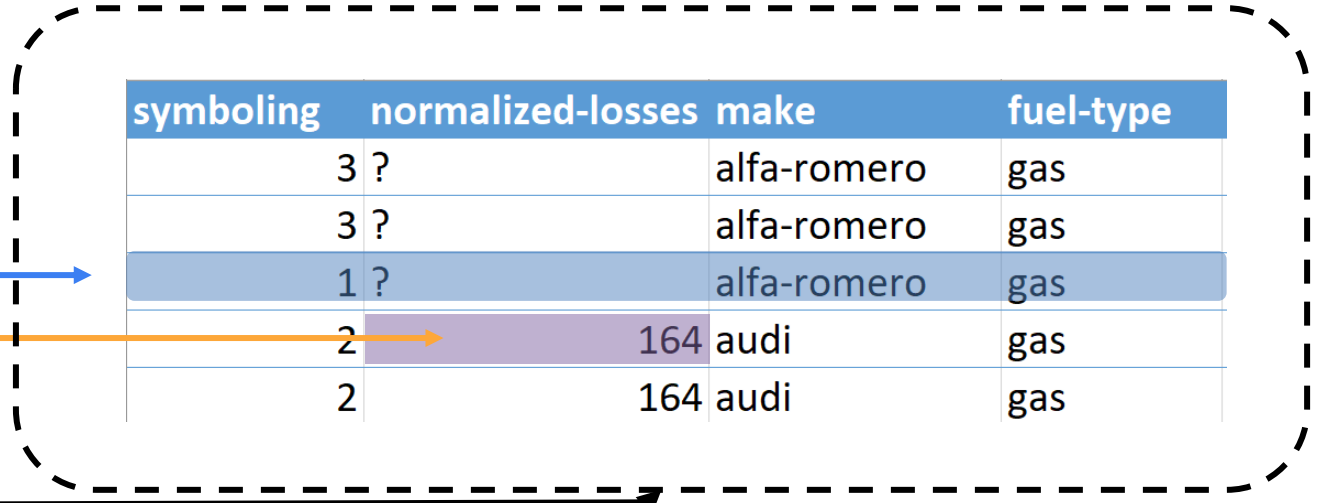
Susunan data

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 takterstruktur (unstructured data)
Sifat	• Model data terdefinisikan sebelumnya • Format butir data (biasanya) teks. • Antar butir data terbedakan dengan jelas. • Ekstraksi/kueri langsung cukup mudah.	• Model data tidak terdefinisikan sebelumnya • Format 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, *time series*	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.

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 aritmetis	Modus, median, rerata aritmetik, rerata geometrik, rerata harmonik
Representasi sebaran	Grouping	Grouping, rentang (*range*), rentang antarkuartil	Grouping, rentang (*range*), rentang antarkuartil, varians, simpangan baku	Grouping, rentang (*range*), rentang antarkuartil, varians, simpangan baku, koefisien variasi
Memiliki nol sejati yang menyatakan nilai mutlak terbawah.	Tidak	Tidak	Tidak	Ya

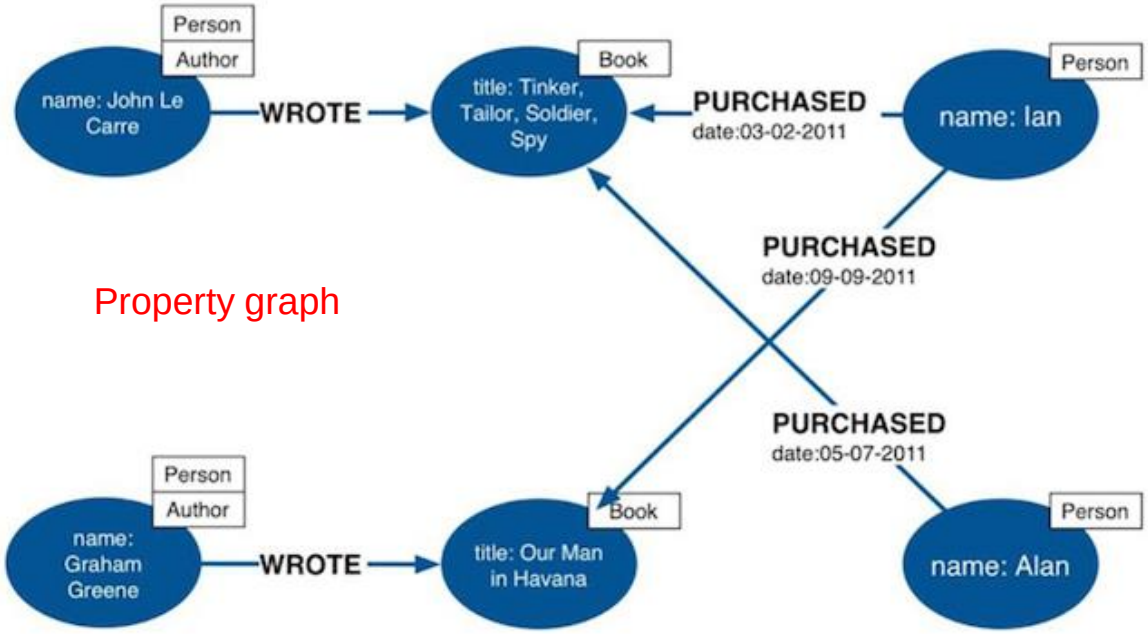
Contoh model data: Tabular

- Terdiri dari N buah rekord (*record*)
- Masing-masing rekord mengandung D buah atribut
- Rekord = baris, *data point*, instans, *example*, transaksi, tupel, entitas, objek, vector fitur.
- Atribut = kolom, *field*, dimensi, fitur.
- Atribut yang sama untuk setiap rekord 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).

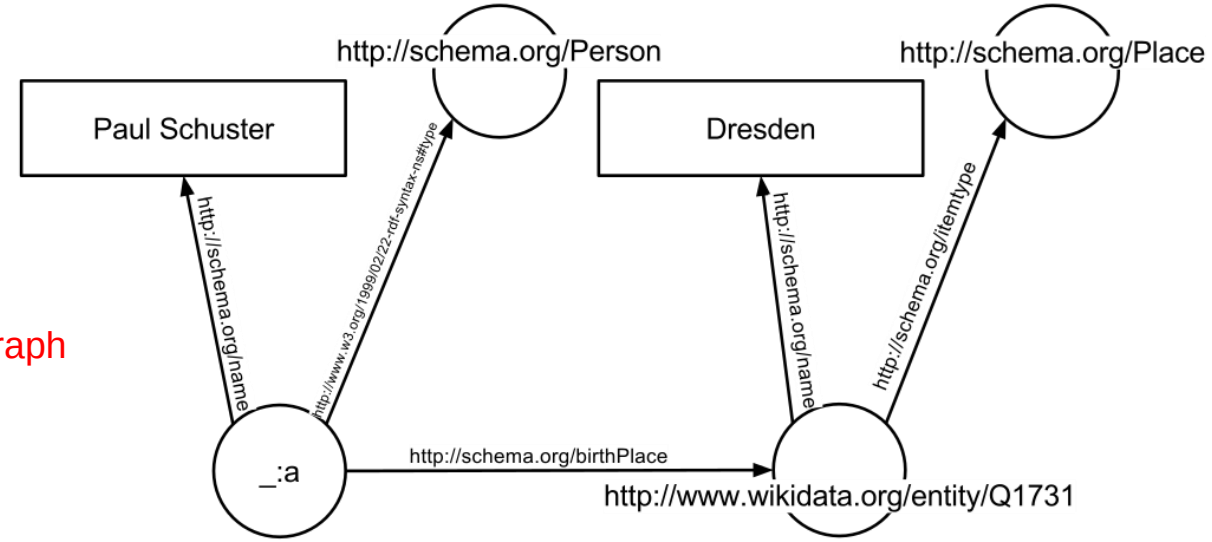
symboling	normalized-losses	make
3	?	alfa-romero
3	?	alfa-romero
1	?	alfa-romero
2	164	audi
2	164	audi

Contoh model data: Graf/Jejaring

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



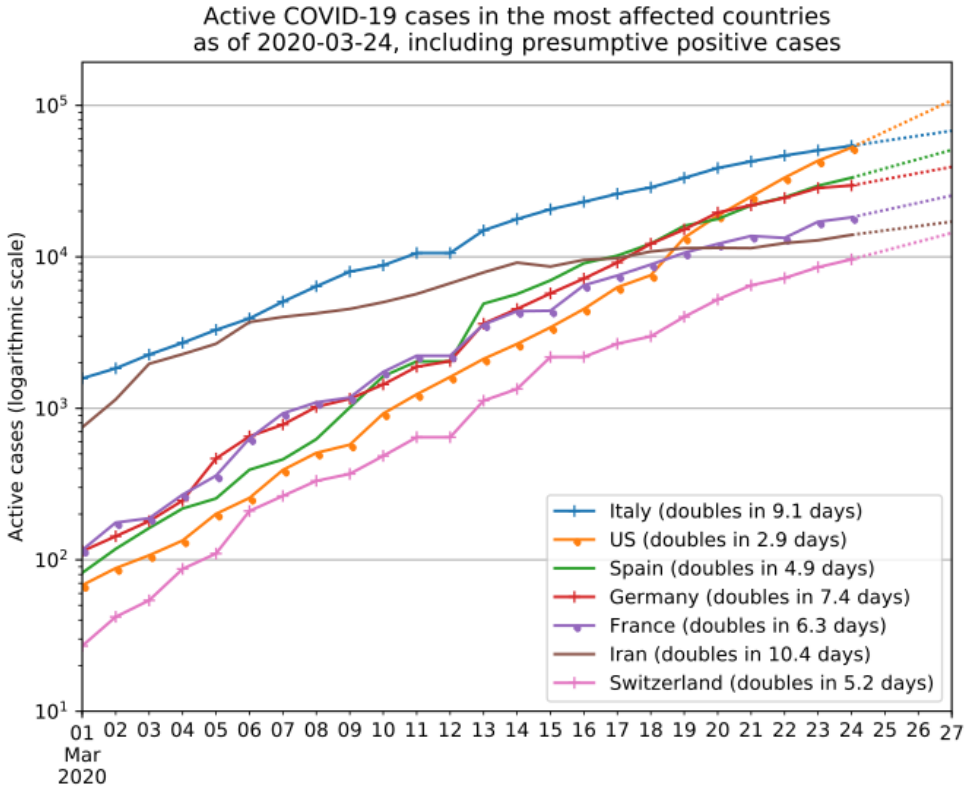
Property graph



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 rekordnya 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

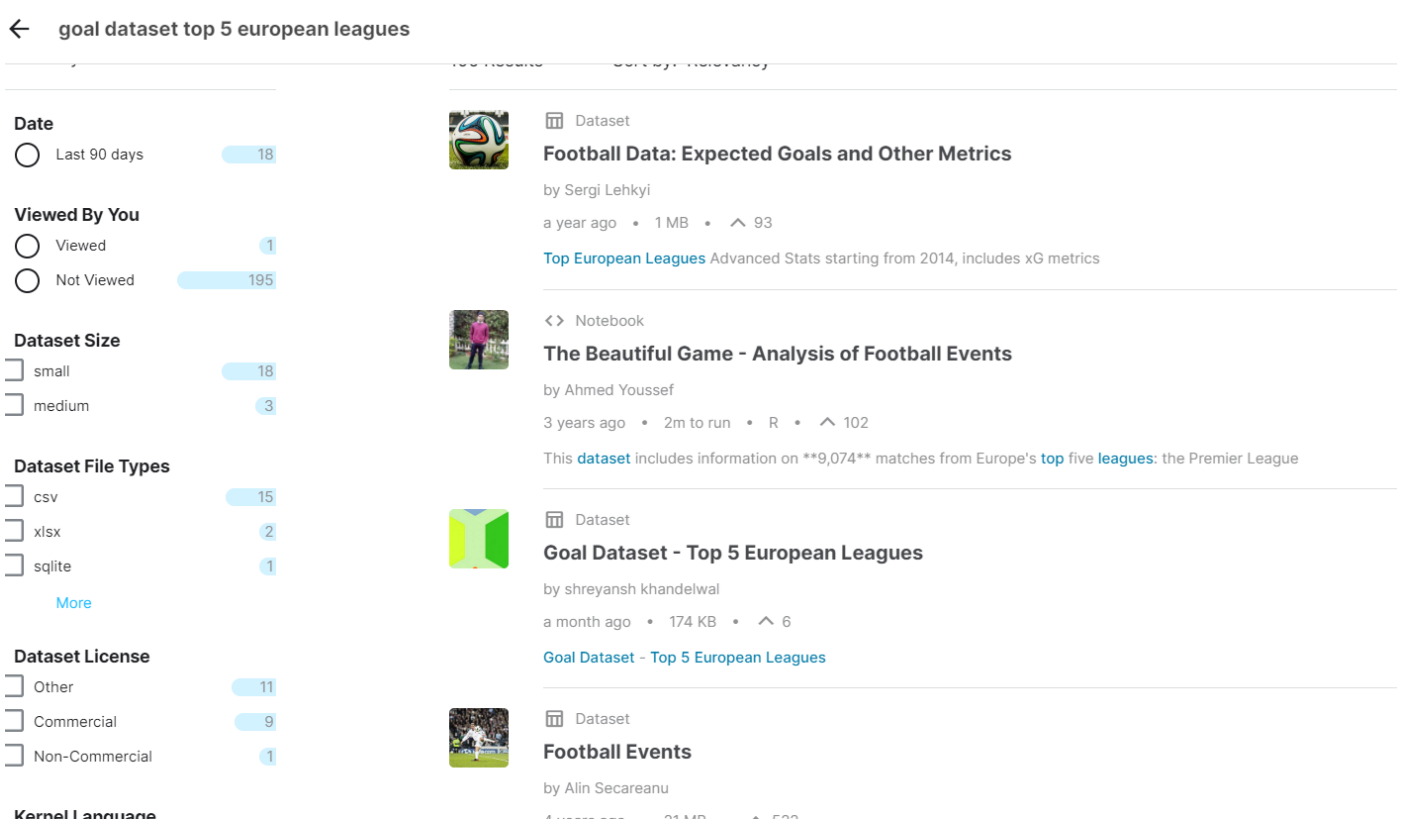
Unduh/salin data ke *local machine*

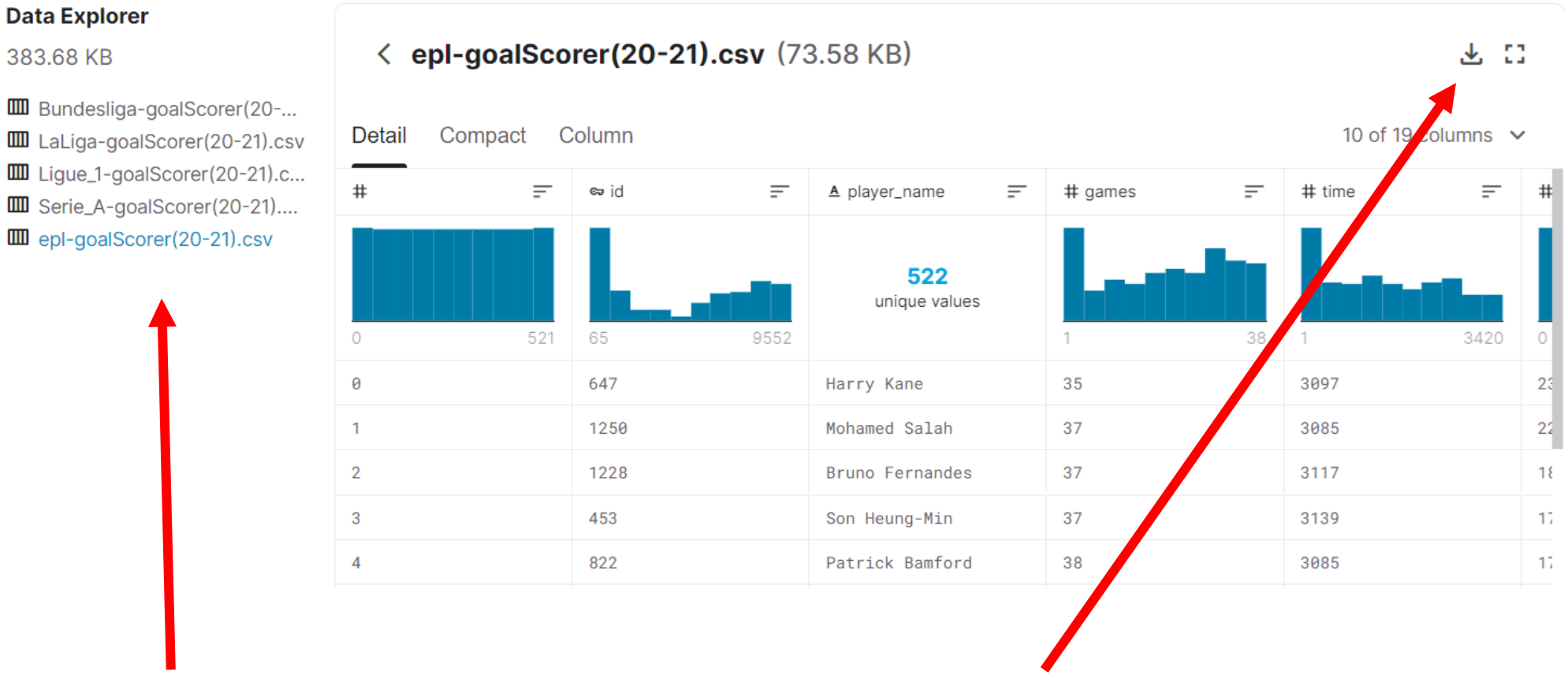
Muat (*load*) data ke pengolah data

- 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"

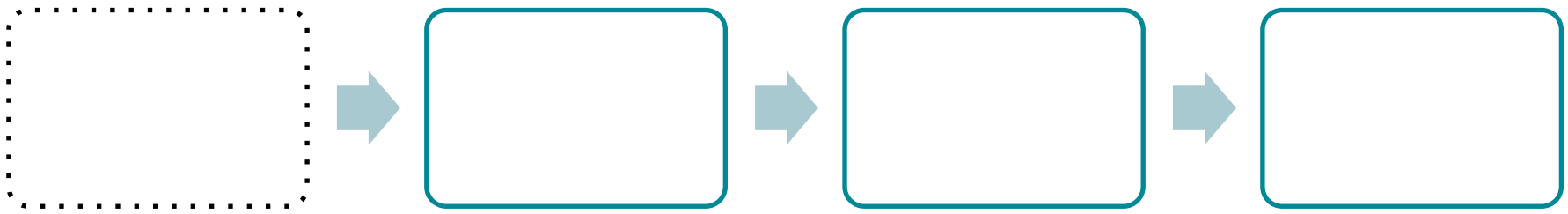




- Di halaman data explorer, pilih "epl-goalScorer (20-21).csv"
- Unduh data dengan mengklik tombol unduh di bagian kanan dan simpan di folder kerja Anda.

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) diperinci di dokumentasi masing-masing layanan.



Mengambil data dengan API dari Kaggle (1)

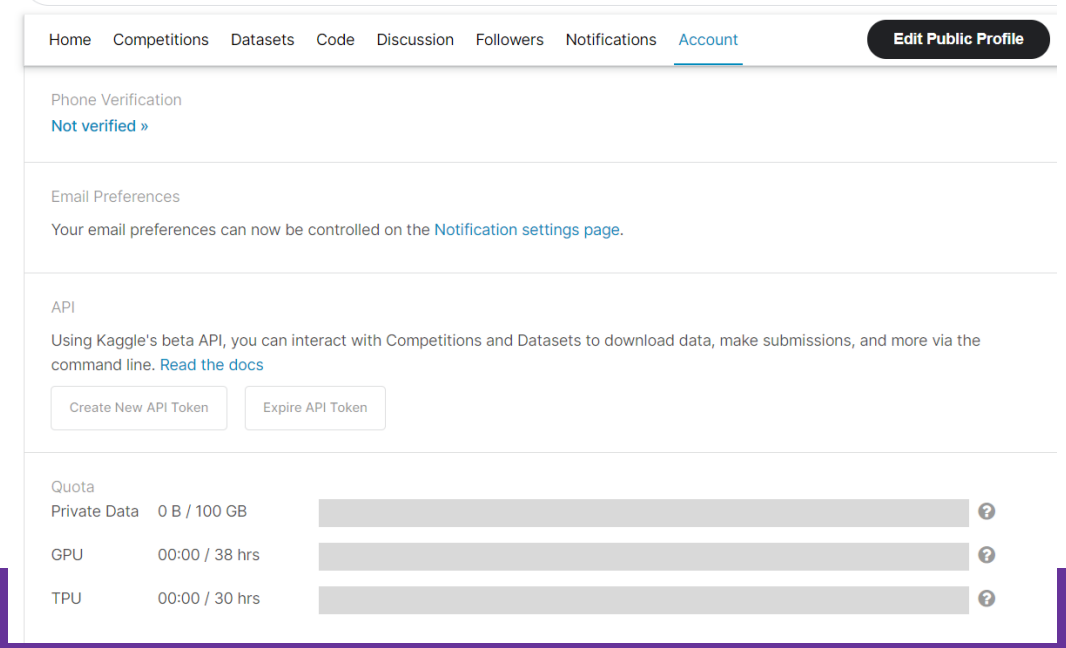
- Nyalakan Jupyter Notebook di folder kerja Anda, lalu buka atau buat satu skrip baru (Python 3).

- 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'



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 `shreyanshkhandelwal/goal-dataset-top-5-european-leagues`.

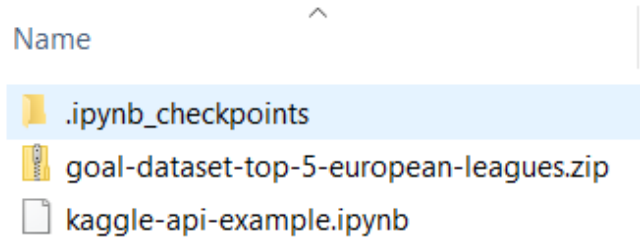
```
In [2]: !kaggle datasets list -s "goal leagues"

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

Mengambil data dengan API dari Kaggle (6)

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

```
In [3]: !kaggle datasets download shreyanshkhandelwal/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.

```
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
```

- Berkas csv dapat langsung dimuat ke Pandas DataFrame

Memuat Data ke Pandas

Memuat data ke Pandas (1)

- Nyalakan Jupyter Notebook di folder kerja Anda.
- Buka atau buat baru satu 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(...)`

```
In [1]: import pandas as pd
        import numpy as np

In [2]: path = "epl-goalScorer(20-21).csv"
        df = pd.read_csv(path)
```

Memuat data ke Pandas (2)

- Method head() dan tail() pada DataFrame membantu kita menampilkan 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	14
1	1	1250	Mohamed Salah	37	3085	22	20.250847	5
2	2	1228	Bruno Fernandes	37	3117	18	16.019454	12

```
df.head()
```

	Unnamed: 0	id	player_name	games	time	goals	xG	assists
0	0	647	Harry Kane	35	3097	23	22.174859	14
1	1	1250	Mohamed Salah	37	3085	22	20.250847	5
2	2	1228	Bruno Fernandes	37	3117	18	16.019454	12
3	3	453	Son Heung-Min	37	3139	17	11.023287	10
4	4	822	Patrick Bamford	38	3085	17	18.401863	7

Telaah Data

Mengungkap tipe-tipe data dari setiap kolom

- Atribut `dtypes` pada DataFrame berisi tipe data dari setiap kolom.
- Lihat Pandas User Guide untuk detil 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).

```
print(df.dtypes)
```

```
Unnamed: 0         int64
id                 int64
player_name       object
games              int64
time               int64
goals              int64
xG               float64
assists            int64
xA               float64
shots              int64
key_passes         int64
yellow_cards       int64
red_cards          int64
position          object
team_title        object
npg                int64
npxG             float64
xGChain          float64
xGBuildup        float64
dtype: object
```

Mengungkap tipe-tipe data dari setiap kolom

- Dua kolom pertama hanyalah ID numerik yang biasanya tidak memiliki makna riil
- Jadi, dari DataFrame `df`, cukup diambil mulai dari kolom "player_name" (untuk *zero-based index*, kita pakai kolom ke-2 dst).

```
df_noid = df.iloc[:,2:]
df_noid
```

	player_name	games	time	goals	xG	assists	xA
0	Harry Kane	35	3097	23	22.174859	14	7.577094
1	Mohamed Salah	37	3085	22	20.250847	5	6.528526
2	Bruno Fernandes	37	3117	18	16.019454	12	11.474996
3	Son Heung-Min	37	3139	17	11.023287	10	9.512992
4	Patrick Bamford	38	3085	17	18.401863	7	3.782247
...	...	...	...	...	...	...	...
517	Jaden Philogene-Bidace	1	1	0	0.000000	0	0.000000
518	Gaetano Berardi	2	113	0	0.074761	0	0.000000
519	Anthony Elanga	1	67	0	0.000000	0	0.000000
520	Femi Seriki	1	1	0	0.000000	0	0.000000

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	522.000000	522.
mean	19.643678	1420.068966	1.862069	2.000806	1.289272	1.376029	17.379310	12.963602	2.061303	0.091954	1.668582	1.
std	11.619836	1031.604819	3.338851	3.317946	2.083350	1.886510	21.572664	16.164361	2.203661	0.295800	2.909929	2
min	1.000000	1.000000	0.000000	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	0.
50%	21.000000	1342.000000	1.000000	0.737295	0.000000	0.691122	10.000000	7.000000	2.000000	0.000000	0.500000	0.
75%	30.000000	2319.000000	2.000000	2.053378	2.000000	2.050509	23.750000	19.000000	3.000000	0.000000	2.000000	1.
max	38.000000	3420.000000	23.000000	22.174859	14.000000	11.474996	138.000000	95.000000	12.000000	2.000000	19.000000	19.

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

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

- Kuadrat dari σ_S, yakni σ_S^2 disebut sebagai **varian**
- 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')
```

	player_name	games	time	goals	xG	assists	xA	sh
count	522	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000
unique	522	NaN	NaN	NaN	NaN	NaN	NaN	N
top	Joel Ward	NaN	NaN	NaN	NaN	NaN	NaN	N
freq	1	NaN	NaN	NaN	NaN	NaN	NaN	N
mean	NaN	19.643678	1420.068966	1.862069	2.000806	1.289272	1.376029	17.379
std	NaN	11.619836	1031.604819	3.338851	3.317946	2.083350	1.886510	21.572
min	NaN	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	NaN	10.000000	470.250000	0.000000	0.074668	0.000000	0.049245	2.000
50%	NaN	21.000000	1342.000000	1.000000	0.737295	0.000000	0.691122	10.000
75%	NaN	30.000000	2319.000000	2.000000	2.053378	2.000000	2.050509	23.750
max	NaN	38.000000	3420.000000	23.000000	22.174859	14.000000	11.474996	138.000

ards	position	team_title	npg	npxG	xGChain	xGBuildup
0000	522	522	522.000000	522.000000	522.000000	522.000000
NaN	14	28	NaN	NaN	NaN	NaN
NaN	M S	Everton	NaN	NaN	NaN	NaN
NaN	106	28	NaN	NaN	NaN	NaN
1954	NaN	NaN	1.668582	1.821450	5.663368	3.455060
5800	NaN	NaN	2.909929	2.931176	5.600249	3.376584
0000	NaN	NaN	0.000000	0.000000	0.000000	0.000000
0000	NaN	NaN	0.000000	0.074668	1.191391	0.720353
0000	NaN	NaN	0.500000	0.715585	4.252738	2.656397
0000	NaN	NaN	2.000000	1.945799	8.308002	5.254647
0000	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 KaneMohamed SalahBruno FernandesSon 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
position        FF M SM SF M SF SF SF SFM SF M SF SF SF SF SF ...
team_title      TottenhamLiverpoolManchester UnitedTottenhamLe...
npg                                                           871
npxG                                                     950.7971
xGChain                                               2956.278233
xGBuildup                                             1803.541131
dtype: object
```

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
```

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
         Name: team_title, dtype: int64
```

Analisa 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
Sheffield United                 1.467599
Southampton                      3.141941
Tottenham                        5.855135
West Bromwich Albion             2.310260
West Bromwich Albion,West Ham         NaN
West Ham                         3.369240
Wolverhampton Wanderers          1.648620
Name: goals, dtype: float64
```

```
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
Sheffield United                 0.666667
Southampton                      1.555556
Tottenham                        2.750000
West Bromwich Albion             1.178571
West Bromwich Albion,West Ham    0.000000
West Ham                         2.478261
Wolverhampton Wanderers          1.222222
Name: goals, dtype: float64
```

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	xGBu
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	0.6
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	0.7
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	0.2
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	0.2
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	0.4
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	0.5
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	0.4
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	0.6
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	0.5
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	0.1
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	0.2
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	0.2
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	0.8
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	1.0

Terima Kasih