



DIGITAL
TALENT
SCHOLARSHIP

TA
Thematic
Academy

Modul Pelatihan Data Understanding 1

Thematic Academy
Digital Talent Scholarship
Tahun 2021

Tujuan Pembelajaran

A. Tujuan Umum

Setelah mempelajari modul ini, peserta latih diharapkan mampu melakukan:

- Pengambilan data untuk proses sains data, dan
- Telaah data dari yang sudah diperoleh dengan menggunakan beberapa metode statistika.

B. Tujuan Khusus

Setelah mempelajari modul ini, peserta latih diharapkan akan mampu:

- Mengakses beberapa sumber data terbuka yang dapat dipakai untuk proses sains data;
- Mendapatkan dataset yang dibutuhkan dari sumber-sumber data tersebut baik secara manual maupun menggunakan library Python, yakni Pandas;
- Mengimpor data ke dalam struktur data Pandas Dataframe;
- Mengekspor data dari Pandas Dataframe ke dalam berkas berformat tabular (CSV, XLS) maupun JSON;
- Mengakses dan memahami atribut-atribut data beserta tipe-tipe datanya yang termuat dalam Pandas Dataframe
- Mengakses dan memahami ringkasan statistik deskriptif dari data pada setiap atributnya;
- Mengelompokkan data menjadi beberapa grup berdasarkan atribut tertentu pada data;
- Melakukan analisis korelasi pada data.

Latar Belakang

Unit kompetensi ini dinilai berdasarkan tingkat kemampuan dalam melakukan pengambilan data dari sumber-sumber yang terbuka dan melakukan telaah atas data tersebut dengan memanfaatkan Pandas yang merupakan sebuah library dalam bahasa pemrograman Python yang umum dipakai untuk sains data.

Deskripsi Pelatihan

Materi ini berisi penjelasan mengenai teknik pengambilan data dari sumber-sumber data terbuka serta cara-cara melakukan telaah data awal yang bersifat eksploratif menggunakan *library* Pandas dari bahasa pemrograman Python. Peserta latihan diasumsikan sudah memiliki kemampuan untuk menjalankan script Python sederhana baik di dalam *shell* ataupun di dalam Jupyter Notebook.

Kompetensi Dasar

- Mampu melakukan pengambilan data untuk proses sains data.
- Mampu melakukan telaah data dengan beberapa metode statistika.

Indikator Hasil Belajar

Dapat melakukan pengambilan data untuk proses sains data dan melakukan telaah data dengan beberapa metode statistika.

INFORMASI PELATIHAN

Akademi	Thematic Academy
Mitra Pelatihan	Kementerian Komunikasi dan Informatika
Tema Pelatihan	Data Scientist: Artificial Intelligence untuk Dosen dan Instruktur
Sertifikasi	<ul style="list-style-type: none">• <i>Certificate of Attainment</i>;• Sertifikat Kompetensi Associate Data Scientist
Persyaratan Sarana Peserta/spesifikasi device/tools/ media ajar yang akan digunakan	Memiliki laptop/komputer dengan spesifikasi minimal : <ul style="list-style-type: none">• RAM minimal 2 GB (disarankan 4 GB)• Laptop dengan 32/64-bit processor

	<ul style="list-style-type: none"> • Laptop dengan Operating System Windows 7, 8, 10, MacOS X atau Linux • Laptop dengan konektivitas WiFi dan memiliki Webcam • Akses Internet Dedicated 126 kbps per peserta per perangkat • Memiliki aplikasi Zoom • Memiliki akun Google Colab
Aplikasi yang akan d gunakan selama pelatihan	<ul style="list-style-type: none"> • Peramban web (<i>web browser</i>) • Spyder • Jupyter notebook
Tim Penyusun	Adila Alfa Krisnadhi, Ph.D.

INFORMASI PEMBELAJARAN

Unit Kompetensi	Materi pembelajaran	Kegiatan pembelajaran	Durasi Pelatihan	Rasio Praktek : Teori	Sumber pembelajaran
Mengumpulkan Data & Menelaah Data	Data Understanding	Daring/Online	Live Class 2 JP LMS 4 JP @ 45 menit	70:30	LMS

Materi Pokok
<ol style="list-style-type: none"> 1. Metodologi Sains Data: Ringkasan 2. Apa itu <i>Data Understanding</i> 3. Sumber Data 4. Susunan Data 5. Tipe dan Model Data <ol style="list-style-type: none"> 5.1. Tipe Data Dasar <ol style="list-style-type: none"> 5.1.1. Data Nominal atau Kategorikal 5.1.2. Data Ordinal 5.1.3. Data Interval

5.1.4. Data Rasio

5.2. Model Data

5.2.1. Model Data Tabular

5.2.2. Model Data Jejaring (*Network*) atau Graf

5.2.3. Model Data Sekuens

6. Pengambilan Data

6.1. Pengambilan Data secara Manual Mengunduh dari Repositori

6.2. Pengambilan Data melalui API

6.2.1. Contoh Pengambilan Data Menggunakan Kaggle API

6.2.2. Contoh Pengambilan Data Menggunakan API di Portal Data Bandung

6.3. Pengambilan Data dengan Web Scraping

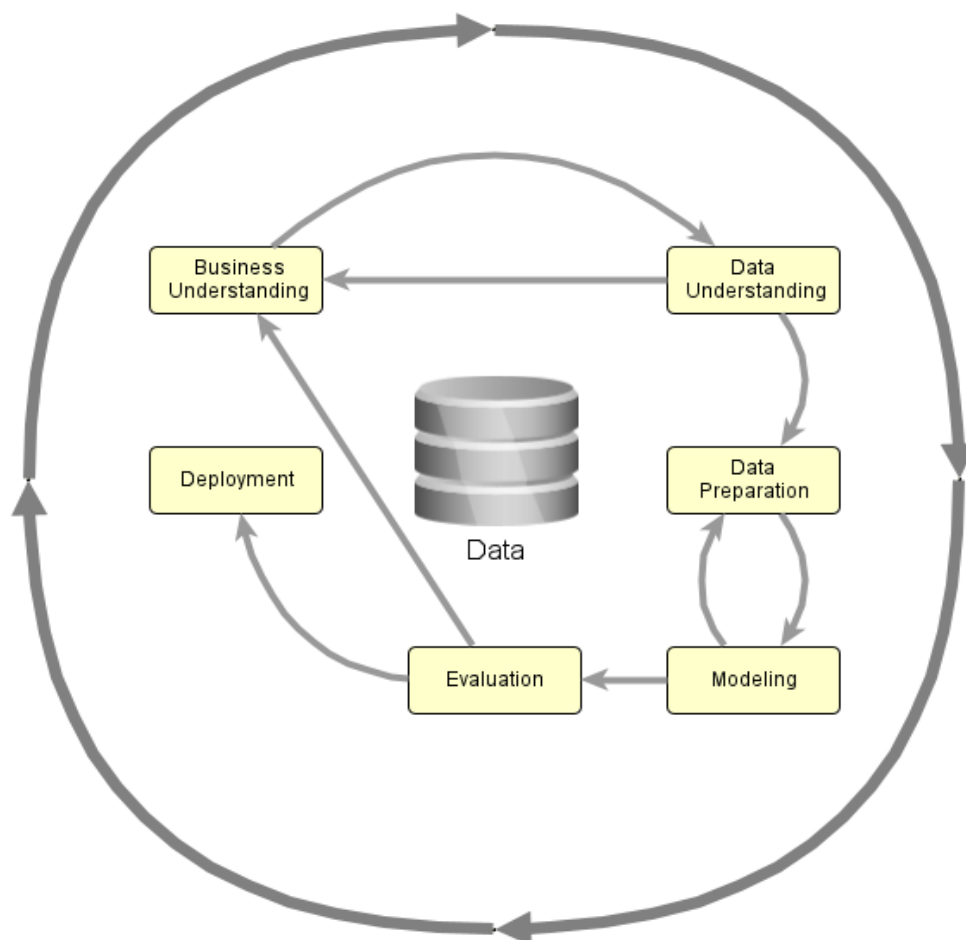
6.4. Pengambilan Data dari Basis Data Relasional

7. Telaah Data

MATERI PELATIHAN

1. Metodologi Sains Data: Ringkasan

Penerapan *artificial intelligence* (AI) di dunia nyata sering diletakkan dalam kerangka metodologi sains data yang juga lazim diadopsi sebagai langkah-langkah pengembangan solusi AI. Metodologi sains data tersebut bertujuan untuk secara sistematis mengekstraksi pengetahuan yang bermanfaat bagi pemecahan permasalahan-permasalahan bisnis yang dihadapi. Formulasi metodologi sains data mengadopsi *Cross-Industry Standard Process for Data Mining* (CRISP-DM) yang dapat dinyatakan dalam beberapa langkah yang tergambar pada Gambar 1. Secara umum, keseluruhan metodologi terdiri dari serangkaian proses yang bersifat iteratif. Penjelasan secara lebih terperinci mengenai metodologi sains data secara keseluruhan dapat dilihat pada modul



Gambar 1. Metodologi sains data menurut CRISP-DM

“Metodologi Pengembangan AI Menggunakan Data”. Sementara itu, modul ini berfokus pada tahapan *data understanding* (pemahaman data).

Tahapan *data understanding* dilakukan setelah permasalahan bisnis yang mendasari pengembangan solusi AI didefinisikan dengan jelas pada tahapan *business understanding* yang rinciannya dibahas di modul “Business Understanding”. Di dalam tahapan *data understanding*, aktivitas pengambilan data serta telaah data dilakukan dengan tujuan mendapatkan gambaran yang utuh atas data yang dapat diperoleh sebagai bahan untuk memecahkan permasalahan bisnis tersebut. Dalam proses mendapatkan gambaran utuh tersebut, dapat saja didapati bahwa formulasi permasalahan bisnis yang dipakai masih kurang tepat. Oleh sebab itu, hasil tahapan *data understanding* bisa jadi mengharuskan pengulangan tahapan *business understanding* untuk merevisi definisi atau ruang lingkup permasalahan bisnis tersebut.

2. Apa itu Data Understanding?

Data understanding adalah sebuah tahapan di dalam metodologi sains data dan pengembangan AI yang bertujuan untuk mendapatkan pemahaman awal mengenai data yang dibutuhkan untuk memecahkan permasalahan bisnis yang diberikan. Permasalahan bisnis yang terdefinisi dengan baik berperan sebagai dasar untuk menentukan data apa saja yang dibutuhkan. Apabila solusi AI dikembangkan demi memecahkan permasalahan bisnis tersebut, maka data dapat dianalogikan sebagai bahan mentah yang diperlukan untuk membangun solusi AI tersebut.

Data biasanya tersedia dalam bentuk kumpulan-kumpulan data yang dapat berada dalam keadaan terpisah-pisah, ataupun terintegrasi secara ketat dan rumit. Masing-masing kumpulan data tersebut pada awalnya dibuat dengan maksud dan tujuan tertentu yang tidak hanya berbeda satu sama lainnya, namun juga dapat berbeda dengan tujuan pengembangan yang tersirat dari definisi permasalahan bisnis yang dihadapi. Oleh sebab itu, setiap kumpulan data tersebut memiliki kekuatan dan batasannya masing-masing serta tingkat kesesuaian yang berbeda-beda dengan permasalahan bisnis yang akan dipecahkan. Akibatnya, pengembangan solusi AI untuk permasalahan bisnis tersebut mungkin saja membutuhkan perpaduan beberapa kumpulan data dari sumber-sumber yang berbeda.

Sebagai contoh, data yang berada di dalam basis data pelanggan, basis data transaksi, dan basis data pemasaran dari sebuah perusahaan yang sama akan mencakup aspek informasi yang berbeda atas sebuah populasi pelanggan yang sama. Tidak hanya itu, masing-masing sumber data tersebut akan memiliki tingkat kekayaan (*richness*) data

serta keandalan (*reliability*) data yang berbeda-beda pula. Di sini, tahapan *data understanding* dimaksudkan untuk memberikan gambaran awal tentang kekuatan dan batasan data serta tingkat kesesuaiannya dengan permasalahan bisnis sebelum langkah-langkah pengembangan solusi AI yang lebih lanjut dapat dilakukan dengan baik.

Di samping untuk mendapatkan gambaran awal mengenai data, tahapan *data understanding* juga berguna untuk mengetahui ketersediaan data. Ada data yang tersedia secara terbuka dan bebas. Ada pula data yang untuk mendapatkannya membutuhkan biaya (uang) atau penyiapan sumber daya dan usaha tersendiri. Bahkan, ada data yang justru belum tersedia pada saat dibutuhkan untuk pengembangan solusi AI sehingga perlu diselenggarakan suatu proyek pembuatan atau pengumpulan data tersendiri untuk mendapatkannya.

Secara umum, tahapan *data understanding* dimulai dengan masukan berupa definisi permasalahan bisnis yang jelas beserta deskripsi proses bisnis yang relevan dengan permasalahan bisnis tersebut. Dari masukan ini, beberapa langkah berikut dapat dilakukan secara paralel jika memungkinkan.

Langkah Pertama adalah **identifikasi bagian-bagian di dalam proses bisnis yang mana data** (yang sudah ada ataupun belum) **dapat berpengaruh terhadap jalannya proses bisnis** tersebut. Di samping itu, pengetahuan di dalam organisasi (baik dari individu-individu yang menguasai maupun dari suatu sistem manajemen pengetahuan) yang relevan terhadap bagian-bagian proses bisnis tersebut juga diidentifikasi. Hal ini akan membantu penentuan ruang lingkup pengembangan solusi AI dan sains data yang dilakukan.

Langkah Kedua adalah **menentukan sumber-sumber data internal dan eksternal organisasi, mekanisme aksesnya, beserta hal-hal lain yang dapat membantu atau justru menghalangi diperolehnya data tersebut**. Hal ini dilakukan pada setiap bagian dalam proses bisnis yang mana data mutlak dibutuhkan,. Di samping itu, perlu ditentukan pula apakah data yang dibutuhkan tidak dapat diperoleh atau tidak tersedia di sumber manapun.

Langkah Ketiga adalah berupa **asesmen (*assessment*) pada setiap kumpulan data** yang ditentukan di atas **untuk menentukan nilai tambah bisnis yang dapat diraih** apabila solusi AI dan sains data dapat direalisasikan dengan data tersebut. Pada langkah

ini, perlu diputuskan untuk data yang tidak tersedia atau tidak dapat diperoleh dari sumber manapun apakah usaha pengumpulan data yang terpisah perlu dilakukan, misalnya survei lapangan atau simulasi tambahan.

Langkah Keempat adalah **mengidentifikasi data lain baik dari sumber internal maupun eksternal organisasi yang dapat membawa perbaikan pada proses bisnis melalui solusi AI yang dibangun**. Data lain yang dimaksud di sini merupakan data tambahan di luar data yang mutlak dibutuhkan untuk mendapatkan solusi AI untuk permasalahan bisnis di atas. Data-data tambahan ini kemudian dapat pula dikumpulkan dan dijadikan bahan pengembangan solusi AI tersebut apabila manfaat penggunaannya melebihi biaya dan usaha yang harus dikeluarkan untuk mendapatkannya.

Realisasi keempat langkah di atas membutuhkan penguasaan teknik-teknik *pengambilan data* serta *telaah data*. Langkah pertama, kedua, dan keempat banyak melibatkan teknik-teknik pengambilan data, sementara langkah ketiga dapat direalisasikan dengan bantuan teknik-teknik telaah data. Teknik-teknik telaah data menggunakan metode-metode statistika serta visualisasi. Pada modul ini, pembahasan teknik telaah data dibatasi pada penggunaan metode statistika saja, sedangkan teknik-teknik visualisasi dibahas secara tersendiri pada modul “Visualisasi Data”.

3. Sumber Data

Data dapat diperoleh dari sumber internal maupun eksternal organisasi. Sumber-sumber data internal organisasi dapat berupa, namun tidak terbatas pada, bentuk-bentuk berikut:

- (1) berkas-berkas *spreadsheets* (Excel, *comma-separated values* (CSV), JSON, dll.),
- (2) basis data internal organisasi yang dapat diakses dengan kueri (SQL, atau bahasa kueri lainnya),
- (3) dokumen-dokumen teks internal organisasi,
- (4) berkas-berkas multimedia (audio dan/atau video).

Sumber-sumber data eksternal organisasi dapat berasal dari organisasi-organisasi lain yang membuka akses khusus terhadap data-data yang mereka miliki, atau dari situs-situs Web yang membagi data dalam berbagai format, moda akses, dan lisensi. Sebagian situs-situs Web tersebut merupakan repositori data terbuka (*open data repository*) memberikan akses terhadap datanya yang dapat secara bebas digunakan kembali

(*reuse*), dimodifikasi, dan dibagikan kembali oleh semua orang untuk kebutuhan apapun. Sebagian yang lain menyediakan data di ranah publik (*public domain*) namun aksesnya hanya dapat diperoleh melalui prosedur permintaan (*request*) tertentu dan lisensinya lebih terbatas (bukan *open license*). Data yang tersedia di Web dapat dicari menggunakan layanan Google Dataset Search (<https://datasetsearch.research.google.com>) atau langsung mengakses repositori data yang memiliki data yang diinginkan. Berikut beberapa contoh repositori data terbuka:

- 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/>).

4. Susunan Data

Dalam konteks sains data, *butir data* (*data item*) dipandang sebagai satu satuan terkecil dari data. Sebagai contoh, usia seorang individu tertentu (yang dinyatakan dalam nilai bilangan bulat tertentu) atau nama jalan (yang dinyatakan dalam sebagai *string* atau serangkaian kata dalam bahasa tertentu) merupakan satu butir data. Pada umumnya, satu butir data yang berdiri atau datang secara sendiri tidak memiliki makna yang jelas karena untuk memahami dan menginterpretasikannya, informasi konteks yang dibawa oleh butir data lain ataupun keterangan penjelas dalam bentuk *metadata* (data mengenai data) mutlak dibutuhkan. Oleh sebab itu, data atau *dataset* dapat ditinjau sebagai suatu susunan atau rangkaian butir-butir data yang memberikan suatu makna yang utuh bagi penggunaanya. Walaupun terkait erat, sifat susunan data yang dimaksud di sini harus dibedakan dengan format berkas (*file format*) yang lebih merujuk kepada susunan data

ketika disimpan di suatu sistem komputer. Ditinjau dari sifat susunannya, data dapat dibagi menjadi dua jenis berikut.

1. **Data *terstruktur*** adalah data yang butir-butirnya tersusun secara jelas mengikuti sebuah struktur yang ditentukan oleh suatu *model data* tertentu (*model data* akan dijelaskan di subbab berikutnya). Pada data terstruktur, model data yang dipakai telah diketahui atau ditetapkan sebelum data tersebut dibuat atau dikonstruksi. Dalam data terstruktur, satu butir data dengan butir data lainnya dapat dibedakan dengan jelas. Di samping itu, kejelasan susunan data membuat ekstraksi dan pemrosesan setiap butir data di dalamnya dapat dilakukan secara langsung. Contoh data terstruktur adalah data tabular, data berorientasi objek (*object-oriented data*), *time series*, dan lain-lain.
2. **Data *tidak terstruktur*** adalah data yang butir-butirnya tidak dengan jelas tersusun mengikuti suatu *model data* tertentu. Perbedaan utamanya dengan data terstruktur adalah ketiadaan model data yang ditentukan di awal sebelum data tersebut dibuat. Akibatnya, data mengandung ketidakteraturan dan ambiguitas yang menyulitkan ekstraksi dan pemrosesan butir-butir datanya. Contoh data tidak terstruktur adalah data teks yang terkandung di dalam dokumen teks bebas (*free-text document*), data audio, dan data video.

5. Tipe dan Model Data

Model data menyatakan abstraksi bentuk atau struktur yang mendasari bagaimana butir-butir data diorganisasikan menjadi satu kesatuan makna. Model data juga menentukan bagaimana butir-butir data tersebut berhubungan satu sama lain, serta bagaimana butir-butir data tersebut dihubungkan dengan entitas di dunia nyata. Contohnya, sebuah model data dapat menentukan bahwa data yang merepresentasikan seorang individu mahasiswa terdiri dari beberapa butir data yang masing-masing mewakili nomor pokok mahasiswa, nama, serta program studi yang diambilnya.

Istilah model data sendiri terkadang juga digunakan untuk menyatakan abstraksi dari objek-objek dan relasi-relasi yang relevan pada suatu ranah aplikasi tertentu. Contohnya, model data pada suatu perusahaan *e-commerce* lazimnya mengandung abstraksi yang merepresentasikan konsep pelanggan, produk, serta transaksi pembelian barang. Namun

pada modul ini, model data dipahami sebagai abstraksi yang menentukan struktur dari data.

5.1 Tipe Data Dasar

Sebelum beberapa contoh model data disajikan, pembahasan dimulai dengan taksonomi tipe data dasar, yakni tipe-tipe data yang membentuk masing-masing butir data secara elementer. Tipe-tipe data ini terkait dengan bagaimana proses pengukuran untuk memperoleh data dilakukan. Taksonomi ini pertama kali diajukan oleh Stevens pada tahun 1946.¹ Istilah-istilah statistika yang muncul di bawah akan dijelaskan lebih terperinci di subbab tersendiri. Rangkuman atas sifat-sifat dari tipe data elementer dapat dilihat pada Tabel 1.

5.1.1 Data Nominal atau Kategorikal

Tipe data *nominal* atau *kategorikal* mewakili butir-butir data yang nilainya berasal dari suatu himpunan diskrit tertentu dan tidak mengasumsikan adanya urutan. Misalnya, warna dasar cahaya dapat bernilai merah, hijau atau biru saja. Apabila himpunan asal dari suatu tipe data nominal hanya berisi dua kemungkinan nilai, maka tipe data ini disebut tipe data *biner*, contohnya nilai kebenaran yang terdiri dari benar dan salah saja.

Ukuran yang relevan pada data nominal hanyalah *keanggotaan* (*membership*) saja. Operasi matematika yang bermakna hanyalah operasi kesamaan (=) dan ketidaksamaan (\neq). Sekumpulan data nominal dapat diwakili oleh nilai tipikalnya (*central tendency*) yang dinyatakan dalam *modus* (nilai yang paling sering muncul) dalam kumpulan data nominal tersebut. Di samping itu, data nominal memfasilitasi representasi sebaran data melalui *pengelompokan* (*grouping*).

5.1.2 Data Ordinal

Tipe data ordinal mewakili butir-butir data yang berasal dari suatu himpunan diskrit tertentu dan mengasumsikan adanya urutan. Jadi, perbedaannya dengan tipe data nominal adalah adanya urutan tertentu yang intrinsik pada data ordinal. Contohnya, capaian mahasiswa pada suatu mata kuliah yang dapat dinyatakan dengan nilai huruf A, B, C, D, atau E memiliki tipe data ordinal. Contoh tipe data ordinal yang lain adalah hasil perlombaan lari marathon yang dinyatakan dengan peringkat 1, 2, 3, dst.

¹ Stanley S. Stevens, "On the Theory of Scales of Measurement", Science 103 (2684): 677-680, 1946.

Tabel 1. Rangkuman sifat-sifat dari tipe-tipe data elementer.

	Nominal/ Kategorikal	Ordinal	Interval	Rasio
<i>Sifat himpunan asal</i>	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
<i>Contoh</i>	Warna (merah, hijau, biru)	Nilai huruf mahasiswa (A, B, C, D, E)	Suhu dalam Celcius, tanggal dalam kalender tertentu	Panjang jalan, suhu dalam Kelvin
<i>Ukuran data menyatakan ...</i>	Membership	Membership, comparison	Membership, comparison, difference	Membership, comparison, difference, magnitude
<i>Operasi matematika</i>	$=, \neq$	$=, \neq, <, >$	$=, \neq, <, >, +, -$	$=, \neq, <, >, +, -, \times, \div$
<i>Representasi nilai tipikal</i>	Modus	Modus, median	Modus, median, rerata aritmetis	Modus, median, rerata aritmetik, rerata geometrik, rerata harmonik
<i>Representasi sebaran</i>	Grouping	Grouping, rentang (<i>range</i>), rentang antarkuartil	Grouping, rentang (<i>range</i>), rentang antarkuartil, varian, simpangan baku	Grouping, rentang (<i>range</i>), rentang antarkuartil, varian, simpangan baku, koefisien variasi
<i>Memiliki nol sejati yang menyatakan nilai mutlak terbawah.</i>	Tidak	Tidak	Tidak	Ya

Setiap ukuran dan operasi pada data nominal juga dapat dipakai pada data ordinal. Di samping itu, ukuran yang relevan bagi data ordinal juga mencakup *perbandingan* (*comparison*) atau *tingkatan* (*level*). Operasi matematika yang relevan juga mencakup

operasi *kurang dari* ($<$) dan *lebih dari* ($>$), sehingga data ordinal dapat diurutkan ke dalam sebuah sekuens yang bermakna. Nilai tipikal data ordinal diwakili oleh *median* atau *nilai tengah*-nya. Sementara itu, persebaran data yang terkandung di dalam sekumpulan data ordinal dinyatakan dalam *rentang* (*range*) atau *rentang antarkuartil* (*interquartile range*).

5.1.3 Data Interval

Tipe data *interval* mewakili butir-butir data numerik yang perbedaan antara butir datanya menunjukkan tingkat selisih antara mereka, namun bukan rasio. Contohnya, suhu udara atau benda yang dinyatakan dalam skala Celsius memiliki dua titik sebagai acuan, yakni titik beku dan titik didih air lalu selisih antar mereka dibagi menjadi 100 bagian atau *interval* yang sama. Contoh lain adalah tanggal dalam kalender tertentu, lokasi dalam sistem koordinat Cartesian atau arah haluan yang dinyatakan dalam besaran derajat relatif terhadap arah utara magnet bumi. Data ini tidak menyatakan rasio karena misalnya panasnya suhu 60 derajat Celsius tidak dapat dikatakan sama dengan dua kali panasnya suhu 30 derajat Celsius, atau bahwa operasi perkalian dan pembagian antara dua buah tanggal tidak memiliki makna sama sekali.

Semua ukuran dan operasi yang relevan untuk data ordinal dapat diterapkan pada data interval. Di samping itu, ukuran yang relevan bagi data interval juga mencakup *perbedaan* atau *selisih* (*difference*). Operasi matematika yang juga relevan adalah *penjumlahan* (+) dan *pengurangan* (-) yang memungkinkan data interval diperbandingkan secara kuantitatif dengan suatu standar tertentu. Nilai tipikal data interval diwakili oleh *rata-rata aritmetik* (*arithmetic mean*). Persebaran data dalam sekumpulan data bertipe interval dinyatakan dengan *varian* atau *simpang baku*. Tipe data interval tidak memiliki titik nol sejati sebagai titik asal yang menunjukkan nilai mutlak terbawah dari data. Dengan kata lain, data interval dapat bernilai negatif atau di bawah nol.

5.1.4 Data Rasio

Tipe data *rasio* mewakili butir-butir data numerik yang menyatakan perbandingan (*ratio*) antara besarnya kuantitas kontinu jenis tertentu yang diukur terhadap besarnya kuantitas unit di jenis yang sama. Misalnya, panjang jalan bertipe data rasio karena panjang 1 km itu 1000 kali besarnya kuantitas panjang 1 meter. Kebanyakan besaran di ilmu fisika dan teknik bertipe rasio, seperti massa, durasi, energi, muatan listrik, dsb.

Semua ukuran dan operasi yang relevan bagi data interval juga berlaku bagi data rasio. Di samping itu, ukuran data rasio benar-benar menyatakan *seberapa besarnya* (*magnitude*) atau *seberapa banyaknya* (*amount*). Pada data rasio, operasi matematika *perkalian* (\times) dan *pembagian* (\div) juga berlaku. Nilai tipikal data rasio dapat diwakili oleh *rata-rata geometrik* (*geometric mean*) dan *rata-rata harmonik* (*harmonic mean*). Sebaran data rasio dapat diwakili oleh nilai *koefisien variasi* (*coefficient of variation*) serta *studentized range*. Tidak seperti tipe data interval, tipe data rasio memiliki titik nol sejati sebagai titik asal yang berperan sebagai nilai mutlak terbawah. Tipe data rasio lazimnya tidak memiliki nilai negatif atau di bawah nol.

5.2 Model Data

Pada data terstruktur, butir-butir datanya yang masing-masing memiliki tipe dasar di antara yang disebut atas dapat disusun menjadi suatu bentuk menurut suatu model data tertentu. Contoh-contoh model data adalah sebagai berikut.

5.2.1 Model Data Tabular

Data *tabular* atau *tabel* terdiri dari sekumpulan n buah rekord (*record*) yang masing-masing terdiri dari d buah *atribut*. Tergantung aplikasinya, rekord disebut juga *baris*, *data point*, *instans*, *example*, *transaksi*, *entitas*, *tupel*, *objek*, *vektor fitur*. Sementara itu, atribut disebut juga *kolom*, *field*, *dimensi*, atau *fitur*. Contoh data tabular sederhana ada pada Gambar 2.

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

Gambar 2. Contoh data tabular sederhana.

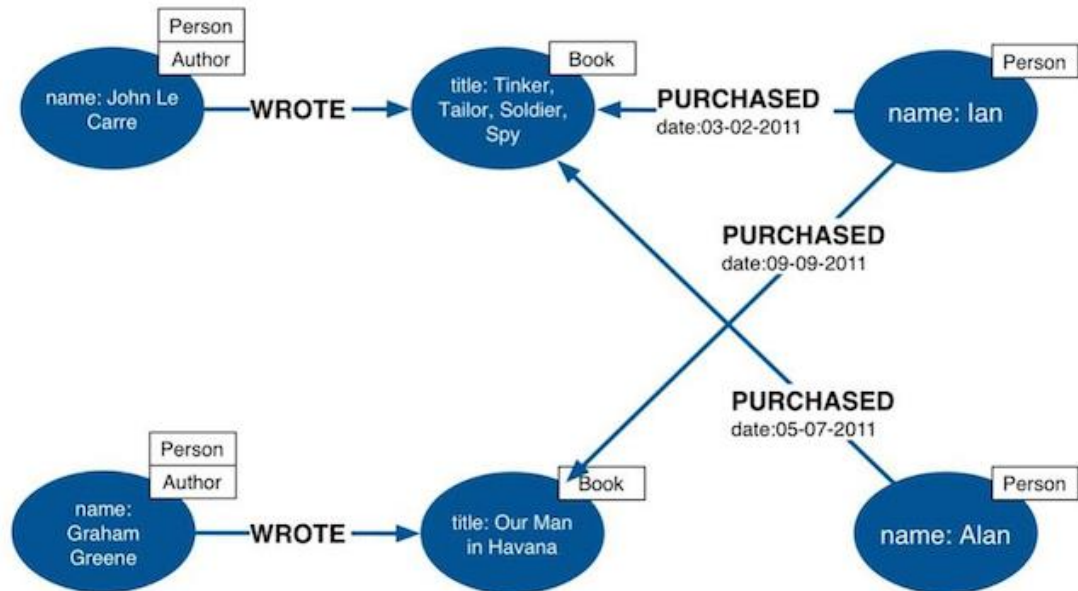
Data tabular merupakan bentuk data yang paling umum ditemukan di dunia nyata. Data tabular tidak mengekspresikan keterhubungan antara satu rekord dengan rekord lainnya, walaupun dalam aplikasinya ketergantungan tersebut dapat diasumsikan ada. Struktur menurut model data tabular lebih menekankan pada hubungan antara masing-masing nilai atribut dari satu rekord yang merepresentasikan satu entitas tertentu. Keseragaman antara satu rekord dengan rekord lainnya muncul dalam bentuk

keseragaman tipe data untuk atribut di posisi yang sama, misalnya jika setiap record mewakili entitas individu manusia, maka atribut usia akan memiliki nilai yang bertipe data rasio. Struktur pada data tabular dapat bersifat ketat (*strict*) seperti pada basis data relasional ataupun tidak seperti pada *spreadsheet* Excel. Model data yang menentukan secara ketat struktur data tabular memungkinkan penggunaan bahasa kueri formal untuk mengekspresikan beragam cara mengakses butir-butir data di dalamnya, misalnya seperti SQL pada basis data relasional.

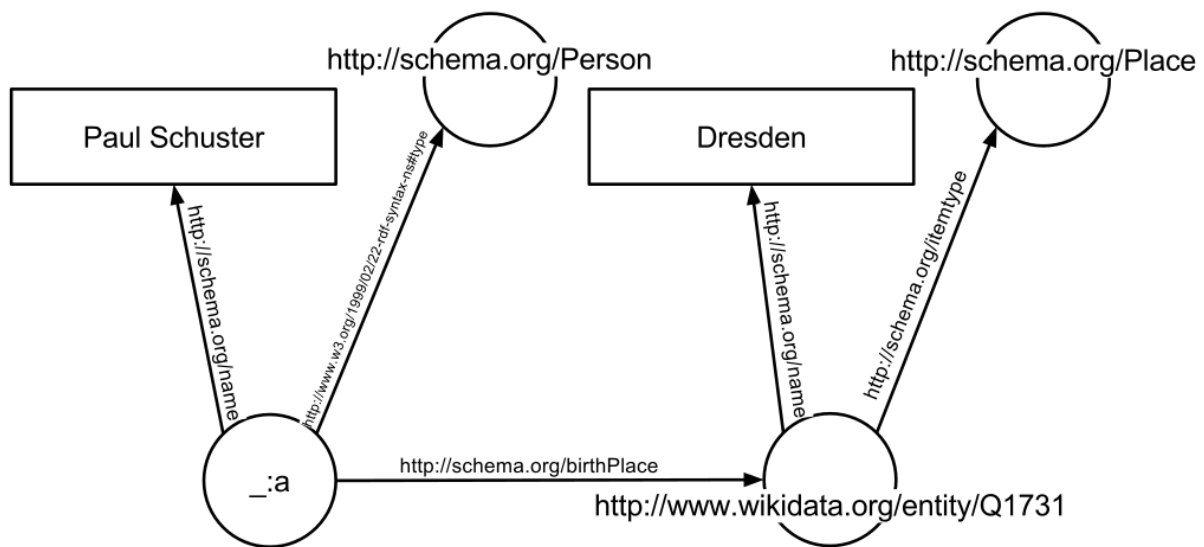
5.2.2 Model Data Jejaring (Network) atau Graf

Data jejaring atau graf terdiri dari sekumpulan *simpul* (*node*) yang masing-masing mewakili satu record yang dapat mengandung sejumlah atribut. Namun, tidak seperti data tabular, antara dua record yang berbeda dapat memiliki jumlah dan jenis atribut yang sama sekali berbeda. Model data graf ini juga dapat mengekspresikan hubungan antara satu record dengan record lainnya secara eksplisit. Varian dari model data graf adalah model data hierarkis (yang mana hubungan antar record merepresentasikan hierarki berbentuk pohon) dan model data berorientasi objek (*object oriented data model*).

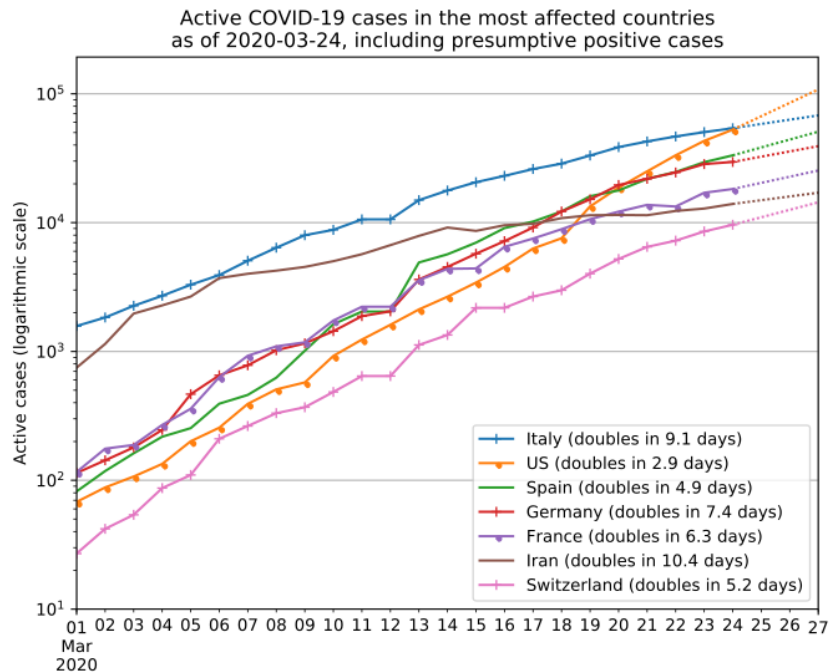
Model data graf modern memiliki dua varian yang paling populer, yakni *property graph* serta *resource description framework* (RDF). Beberapa contoh implementasi basis data graf adalah Neo4j, Apache Tinkerpop, GraphDB, Virtuoso, AllegroGraph, Oracle Spatial and Graph, dll. Basis data graf biasanya memiliki bahasa kueri masing-masing seperti Cypher, Gremlin, GraphQL, atau SPARQL. Sebagian di antara bahasa kueri tersebut telah ditetapkan sebagai standar oleh beberapa badan standar. Model data graf menjadi populer di beberapa tahun terakhir karena digunakan oleh beberapa penyedia layanan media sosial untuk merepresentasikan koneksi sosial. Contoh model data graf berbasis RDF dan *property graph* dapat dilihat pada Gambar 3 dan Gambar 4.



Gambar 3. Model property graph yang menggambarkan jejak pembelian buku oleh dua orang. Sumber: М.Оюунболор, CC BY-SA 4.0 via Wikimedia Commons, https://upload.wikimedia.org/wikipedia/commons/a/a6/Property_graph_model.png



Gambar 4. Contoh model data graph berbasis RDF menggambarkan tokoh Paul Schuster beserta tempat lahirnya. (Sumber: Denny, CC0, via Wikimedia Commons, https://upload.wikimedia.org/wikipedia/commons/0/09/RDF_example.svg)



Gambar 5. Contoh visualisasi dari data sekuens. Sumbu horizontal menyatakan atribut kontekstual, yakni penanda waktu, lalu sumbu vertikal menyatakan nilai atribut behavioral, yakni jumlah kasus aktif COVID-19. Butir-butir data diwakili titik-titik pada grafik. (Sumber: [Pascal Getreuer, CC BY-SA 4.0](https://commons.wikimedia.org/wiki/File:Time_series_of_active_COVID-19_cases_most_affected_countries_as_of_2020-03-21.svg), via Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Time_series_of_active_COVID-19_cases_most_affected_countries_as_of_2020-03-21.svg)

5.2.3 Model Data Sekuens

Model data sekuens terdiri dari rekord-rekord yang terhubung satu sama lain secara sekuensial. Beberapa contoh data sekuens adalah data yang dihasilkan oleh suatu sensor suhu selama suatu rentang waktu; data yang dihasilkan oleh alat elektrokardiogram (ECG) yang mengandung informasi detak jantung seorang pasien yang diukur dalam suatu rentang waktu; data di dalam *event log* yang merekam aktivitas pengguna suatu situs web, atau data nukleotida yang tersusun dalam rangkaian yang membentuk sebuah protein. Rekaman video atau audio juga dapat digolongkan sebagai data sekuens. Gambar 5 memberikan contoh visualisasi data sekuens.

Struktur pada data sekuens tersirat secara implisit dari urutan kemunculan rekord pada data tersebut. Oleh sebab itu, atribut-atribut pada data sekuens dapat digolongkan menjadi atribut *kontekstual* yang mendefinisikan konteks yang menjadi basis dependensi implisit tersebut, serta atribut *behavioral* yang menyatakan butir-butir data yang nilainya diperoleh pada suatu konteks tertentu. Contohnya, pada data dari sensor suhu, informasi *time stamp* merupakan atribut kontekstual, sementara besarnya suhu merupakan atribut

behavioral. Bentuk khusus dari model data sekuens adalah *time series* yang atribut kontekstualnya adalah titik waktu atau *time stamp*.

6. Pengambilan Data

Untuk mengakses data dari sumber-sumber data baik internal maupun eksternal organisasi, terdapat setidaknya empat moda akses, yakni:

- (1) mengakses secara manual melalui unduhan berkas data secara langsung atau memperolehnya via kanal komunikasi tertentu seperti email atau kiriman lewat aplikasi *chat*; atau
- (2) mengakses secara programatik melalui Application Programming Interface (API); atau
- (3) mengakses secara programatik dengan mengekstraksi langsung dari laman Web (*Web scraping*)
- (4) mengakses secara programatik ke basis data relasional (*relational database*) yang ada di dalam organisasi.

Modul ini akan memfokuskan pembahasan pada data tabular yang paling umum ditemui.

6.1 Pengambilan Data secara Manual dengan Mengunduh dari Repositori

Pengambilan data secara manual pada dasarnya mengikuti hanya terdiri dari tiga langkah: pencarian data di sumber data, penyalinan/pengunduhan data ke perangkat kerja/mesin tempat analisa dilakukan, serta pemuatan data ke dalam *library* pengolahan data. Berikut contohnya untuk data yang diperoleh dari Kaggle.

Andaikan data yang akan dianalisa adalah data statistik kompetisi sepakbola Eropa pada musim 2020/2021 yang terdapat dari Kaggle. Login ke layanan Kaggle pada alamat <https://www.kaggle.com> melalui aplikasi peramban web (*web browser*). Buat akun jika perlu. Lalu lakukan pencarian dengan kata kunci yang diinginkan. Alternatifnya, pemilihan *dataset* dapat dilakukan dengan penelusuran melalui menu “Datasets” di halaman muka Kaggle.

Pada bahasan ini, kata kunci pencarian yang dipakai adalah “goal top 5 european leagues”. Hasil pencarian di Kaggle dengan kata kunci pencarian tersebut ditampilkan pada Gambar 6. Kemudian, tautan ke *dataset* yang diinginkan di-klik untuk masuk ke halaman *data explorer* dari *dataset* tersebut. Untuk keperluan modul ini, klik “epl-goalScorer (20-21).csv” di bagian kiri halaman *data explorer* untuk menghasilkan

Gambar 7. Selanjutnya, berkas CSV tersebut dapat diunduh dengan mengklik tautan unduh di sebelah kanan atas. Setelah diunduh, data siap digunakan untuk proses selanjutnya, yakni telaah data. Pemuatan data ke dalam skrip pengolah data yang dibuat menggunakan *library* Pandas dari Python akan dibahas pada Bagian 7.1.

6.2 Pengambilan Data melalui API

Selain didapatkan secara manual, data bisa diperoleh dengan memanfaatkan *Application Programming Interface* (API) publik yang disediakan oleh beberapa layanan data. Beberapa layanan data yang disebutkan pada Bagian 3, seperti Kaggle, Portal Satu Data Indonesia, atau Portal Data Bandung merupakan contoh layanan data yang menyediakan API untuk mengakses datanya. Secara umum, langkah-langkah untuk mengakses data melalui API publik terdiri dari langkah-langkah berikut:

- (1) penyiapan *API token/key*: langkah ini bersifat opsional, tergantung pada ketentuan sistem layanan data yang hendak diakses yang biasanya diperinci di dalam dokumentasi yang diberikan oleh masing-masing layanan;
- (2) akses data ke layanan data dengan melakukan pemanggilan fungsi API yang sesuai;
- (3) pencarian data yang diinginkan memanfaatkan fungsi-fungsi yang diberikan oleh API; dan
- (4) pemuatan data dari layanan data ke skrip pengolah data.

← goal top 5 european leagues

Searching for goal top 5 european leagues within

<> Notebooks 466 Topics 42 Comments 23 Datasets 21

Filter by

552 Results

Sort by: Relevancy

Date

☐ Last 90 days 38
☐ Last week 1

Viewed By You

☐ Viewed 1
☐ Not Viewed 551

Dataset Size

☐ small 18
☐ medium 3

Dataset File Types

☐ csv 15
☐ xlsx 2
☐ sqlite 1
[More](#)



Dataset

Football Data: Expected Goals and Other Metrics

by Sergi Lehkyi

a year ago • 1 MB • 96

[Top European Leagues](#) Advanced Stats starting from 2014, includes xG metri



<> Notebook

The Beautiful Game - Analysis of Football Events

by Ahmed Youssef

4 years ago • 2m to run • R • 105

in Europe's [top 5 leagues](#), x="y=") + theme(axis.text = element_blank(), axis.t



Dataset

Goal Dataset - Top 5 European Leagues

by shreyansh khandelwal

2 months ago • 174 KB • 6

[Goal Dataset - Top 5 European Leagues](#)

Dataset License

Gambar 6. Tampilan hasil pencarian di Kaggle.

Data Explorer

383.68 KB

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

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



Detail Compact Column

10 of 19 columns

#	id	player_name	# games	# time	#
0	521	522 unique values	38	3420	0
8	647	Harry Kane	35	3097	23
1	1250	Mohamed Salah	37	3085	22
2	1228	Bruno Fernandes	37	3117	18
3	453	Son Heung-Min	37	3139	17
4	822	Patrick Bamford	38	3085	17

Gambar 7. Halaman data explorer dari dataset "Goal Dataset - Top 5 European Leagues". Klik "epl-goalScorer (20-21).csv" untuk mendapatkan tampilan ini. Untuk mengunduh berkas CSV tersebut, klik tautan unduh di kanan atas.

6.2.1 Contoh Pengambilan Data Menggunakan Kaggle API

Kaggle (<https://www.kaggle.com>) menyediakan API berbasis Python untuk mengakses data di dalamnya. API ini dapat dijalankan di Jupyter Notebook.

Langkah pertama: Penyiapan notebook

Pertama-tama, Anda perlu **menyiapkan Jupyter Notebook** untuk bekerja dengan Kaggle API. Buka sebuah Jupyter Notebook baru dan lakukan instalasi Kaggle API dengan perintah berikut.

```
!pip install kaggle
```

Langkah kedua: Pembuatan token API

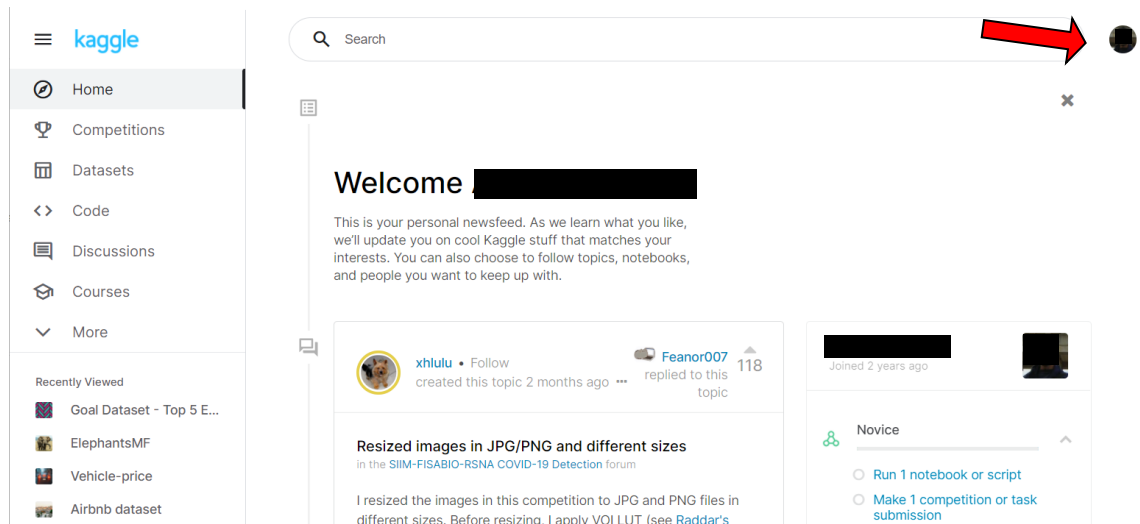
Berikutnya, Anda perlu **membuat token API**. Untuk melakukannya login ke Kaggle dan akses halaman profil Anda dengan meng-klik foto profil Anda yang terletak di sebelah kanan atas (Gambar 8). Kemudian, akan muncul menu seperti pada **Error! Reference source not found..** Halaman profil dapat diakses dengan meng-klik “Your Profile”. Pada halaman profil (Gambar 9), klik menu “Account” dan di bagian tengah agak ke bawah, klik tombol “Create New API Token” (jika gagal, bisa diulang dengan meng-klik “Expire API Token” lebih dahulu). Setelah tombol “Create New API Token” di-klik, peramban akan mengunduh berkas `kaggle.json` ke dalam folder unduhan.

Berkas `kaggle.json` harus ditaruh di folder berikut agar Kaggle API dapat bekerja. Jika folder tersebut belum ada, buat dulu dengan perintah `mkdir` di *shell/command line*.

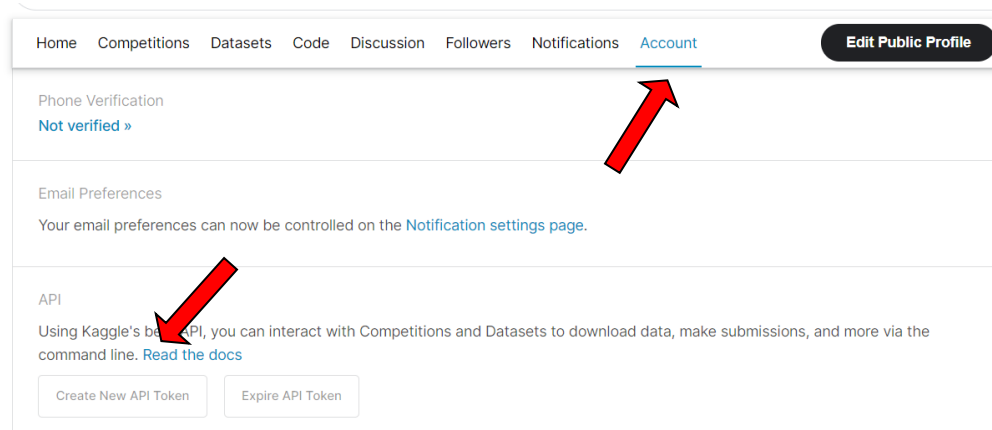
(1) `~/ .kaggle/` (untuk sistem Linux/Mac)

(2) `C:\Users\<Windows-username>\.kaggle\` (untuk sistem Windows).

Pindahkan file `kaggle.json` ke folder tersebut (menggunakan File/Windows Explorer atau melalui perintah `mv` atau `move` di *shell*). Berkas `kaggle.json` berisi *username* Kaggle beserta string *key* yang diasosiasikan dengan *username* tersebut. Oleh sebab itu, pada praktiknya, berkas ini harus diamankan agar tidak diakses oleh pihak yang tidak berhak.



Gambar 8. Halaman muka Kaggle. Akses profil dengan meng-klik foto di kanan atas.



Gambar 9. Halaman profil untuk membuat API token.

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
jacobbaruch/basketball-players-stats-per-season-49-leagues-24 00:09:21	4328	104	1.0	Basketball Players Stats per Season - 49 Leagues	3MB	2020-11
eladsil/football-games-odds-15 08:15:14	609	19	0.8235294	Football Matches Odds	5MB	2018-10
chuckkephron/leagueoflegends-29 18:59:35	9690	274	0.7647059	League of Legends	30MB	2018-01
jashsheth5/indepth-soccer-statistics-xg-xa-and-more-26 05:36:58	177	3	0.9411765	In-depth soccer statistics: xG, xA and more	1MB	2020-09
mohamedhanyyy/top-football-leagues-scorers-04 18:30:38	493	25	1.0	Top Football Leagues Scorers	18KB	2020-12
sashchernuh/european-football-01 13:16:26	4263	160	0.8235294	World Soccer DB: archive of odds [01-JUN-2021]	48MB	2021-06

Gambar 10. Hasil pencarian dataset di Kaggle

```

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

```

Gambar 11. Hasil perintah ekstraksi dataset dari Kaggle.

Langkah ketiga: Pencarian dataset

Setelah token API diperoleh, **pencarian dataset** data dilakukan dengan fungsi-fungsi dalam Kaggle API. Secara umum, Kaggle API menyediakan empat kategori perintah terkait kompetisi yang diselenggarakan Kaggle, *dataset* yang disediakan Kaggle, penggunaan Kaggle Kernels (perkakas pengembangan kolaboratif di Kaggle), serta konfigurasi Kaggle API sendiri. Daftar perintah ditampilkan pada

Tabel 2 dan detail penggunaan semua perintah dapat dilihat di halaman Kaggle API di <https://github.com/Kaggle/kaggle-api>. Pada modul ini, hanya perintah terkait *dataset* saja yang akan digunakan.

Pencarian dataset dapat dilakukan dengan menjalankan perintah pencarian di Jupyter Notebook. Untuk kata kunci pencarian “goal leagues”, perintahnya adalah sebagai berikut.

```
!kaggle datasets list -s “goal leagues”
```

Perintah tersebut akan menghasilkan luaran kurang lebih seperti pada Gambar 10. *Identifier dataset* terdapat di kolom pertama pencarian. Misalnya, *dataset* “Goal Dataset – Top 5 European Leagues” memiliki *identifier* shreyanshkhanelwal/goal-dataset-top-5-european-leagues.

Tabel 2. Daftar fungsi/perintah pada Kaggle API. Yang dipakai pada modul ini hanyalah perintah terkait dataset.

Perintah	Makna Perintah
kaggle competitions list	Daftar kompetisi yang tersedia
kaggle competitions files	Daftar berkas-berkas kompetisi
kaggle competitions download	Unduh berkas-berkas kompetisi
kaggle competitions submit	Kirim satu <i>submission</i> ke kompetisi
kaggle competitions submissions	Lihat semua <i>submission</i> ke kompetisi

kaggle competitions leaderboard	Akses informasi klasemen (<i>leaderboard</i>) kompetisi
kaggle datasets list	Daftar <i>dataset</i> yang tersedia
kaggle datasets files	Daftar berkas-berkas dalam suatu <i>dataset</i>
kaggle datasets download	Unduh berkas-berkas suatu <i>dataset</i>
kaggle datasets create	Buat sebuah <i>dataset</i> baru
kaggle datasets version	Buat versi baru dari suatu <i>dataset</i>
kaggle datasets init	Inisialisasi metadata untuk pembuatan <i>dataset</i>
kaggle kernels list	Daftar instans <i>kernel</i> yang tersedia
kaggle kernels init	Inisialisasi metadata <i>kernel</i>
kaggle kernels push	Kirim kode/skrip ke sebuah <i>kernel</i>
kaggle kernels pull	Ambil kode/skrip dari sebuah <i>kernel</i>
kaggle kernels output	Ambil data luaran dari eksekusi skrip terakhir pada sebuah <i>kernel</i>
kaggle kernels status	Tampilkan status eksekusi skrip terakhir pada sebuah <i>kernel</i>
kaggle config view	Tampilkan detil pengaturan pada API
kaggle config set	Set nilai pengaturan tertentu pada API
kaggle config unset	Hapus nilai pengaturan tertentu pada API

Langkah keempat: Pengunduhan dataset

Setelah *identifier* dataset adalah **mengunduh *dataset* yang diinginkan**. Perintah unduh membutuhkan *identifier* dataset yang sudah diperoleh sebelumnya. *Dataset* dengan *identifier* shreyanshkhandelwal/goal-dataset-top-5-european-leagues dapat diunduh dengan perintah berikut.

```
!kaggle datasets download shreyanshkhandelwal/goal-dataset-top-5-
european-leagues
```

Dataset akan terunduh di folder aktif Jupyter Notebook dalam bentuk berkas berformat kompresi zip. Berkas tersebut lalu dapat diekstraksi dengan perintah berikut yang hasilnya ada pada Gambar 11. Setiap berkas CSV yang dihasilkan siap dimuat ke Pandas.

```
!unzip goal-dataset-top-5-european-leagues.zip
```

```
(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']
```

Gambar 12. Daftar dataset di Portal Data Bandung sebagai hasil perintah API.

6.2.2 Contoh Pengambilan Data Menggunakan API di Portal Data Bandung

Contoh kedua pengambilan data menggunakan API adalah dari Portal Data Bandung. Portal ini merupakan satu contoh portal data terbuka yang disediakan lembaga pemerintahan, misalnya Pemerintah Kota Bandung, Jawa barat. Data di dalamnya dapat diakses secara manual dengan melalui peramban web pada alamat <http://data.bandung.go.id>.

Portal Data Bandung juga menyediakan API untuk mengakses datanya. Portal tersebut dibangun menggunakan *framework* CKAN sehingga API untuk akses data juga menggunakan CKAN API. Dokumentasi lengkap CKAN API dapat diakses di <https://docs.ckan.org>. Modul ini akan memberikan contoh cara mengakses data di Portal Data Bandung menggunakan CKAN API memanfaatkan *library* requests, json, dan tqdm dari Python.

Langkah pertama: Penyiapan notebook

Pertama-tama, Jupyter Notebook baru disiapkan untuk mengakses datanya. Ketiga *library* requests, json, dan tqdm tersebut diimpor dulu ke dalam Notebook.

```
import requests  
import json  
from tqdm import tqdm
```

Langkah kedua: Menampilkan daftar dataset

Selanjutnya, daftar semua *dataset* yang ada di Portal Data Bandung ditampilkan dengan menggunakan perintah CKAN API berikut.

```
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
```

Perintah `requests.get` mengirim HTTP Get *request* ke server Portal Data Bandung. URL yang dijadikan argumen perintah tersebut adalah **alamat spesifik** untuk mendapatkan daftar *dataset* yang ada pada Portal Data Bandung sesuai dengan spesifikasi CKAN API, yakni URL http://data.bandung.go.id/api/3/action/package_list. Keluarannya adalah sebuah objek HTTP *response* yang mengandung beberapa informasi. Untuk pemrosesan data ini, yang dipakai adalah konten di dalam atribut `text` yang berformat JSON. Konten tersebut dimuat ke dalam suatu Python *dictionary* menggunakan perintah `json.loads`.

```
Result count: 47
10
{'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',
```

Gambar 13. Hasil pencarian pada Portal Data Bandung.

Isi *dictionary* `ds_list_result['result']` pada Gambar 12 menunjukkan bahwa terdapat 3014 *dataset* di Portal Data Bandung dengan nama-nama yang terkandung di dalam *list* tersebut. Dari daftar yang diperoleh tersebut, data yang diinginkan dapat

dipilih dengan melakukan filter melalui skrip yang dibuat sendiri. Alternatifnya, CKAN API juga menyediakan perintah untuk melakukan pencarian *dataset*. Untuk Portal Data Bandung, prosesnya juga menggunakan `requests.get` ke http://data.bandung.go.id/api/3/action/package_search menggunakan potongan kode berikut.

```
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
```

Langkah ketiga: Identifikasi nama dataset yang diinginkan


Andaikan *dataset* yang diinginkan adalah yang terkait dengan “sekolah dasar”. Skrip berikut melakukan pencarian dengan kata kunci pencarian “sekolah dasar”. Kata kunci ini diberikan sebagai parameter HTTP dengan nama `q`. Nama parameter ini sesuai konfigurasi CKAN API. Keluaran potongan skrip di atas ada pada Gambar 13. Hasil pencarian pada Portal Data Bandung.. Sebagaimana terlihat pada gambar, seharusnya pencarian menghasilkan 47 *dataset*, namun ternyata `list search_result['result']['results']` yang seharusnya berisi detil dari semua *dataset* tersebut justru hanya berisi detil dari 10 *dataset* saja. Ini adalah pembatasan pada konfigurasi CKAN API yang dipasang oleh Portal Data 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',
  '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'.
```

Gambar 14. Nama-nama dataset yang mengandung "sekolah" dan "dasar". Dataset yang diinginkan adalah "jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah".

Alternatifnya, skrip dibuatkan untuk menyaring nama-nama *dataset* yang diinginkan. Andaikan *dataset* yang diinginkan mengandung setidaknya salah satu kata dari frasa "sekolah dasar". Maka, daftar pada Gambar 12 tersaring menjadi pada Gambar 14.

```
sd_datasets = [ds_name for ds_name in ds_list_result['result'] \
               if 'sekolah' in ds_name and 'dasar' in ds_name]
sd_datasets
```



```

{
  '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': ''
      }
    ]
  }
}

```

Gambar 15. Rincian dataset "jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah"

Langkah keempat: Pengumpulan alamat URL berkas dalam dataset yang diinginkan

Setiap *dataset* di dalam Portal Data Bandung terdiri dari satu atau lebih berkas. Oleh karena itu, setelah nama *dataset* yang diinginkan ditentukan, langkah yang dilakukan bertujuan untuk mendapatkan semua alamat URL dari masing-masing berkas yang terkandung di dalam *dataset* yang tersebut. Andaikan nama dataset yang diinginkan adalah "jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah". Maka, rincian dari *dataset* tersebut dapat diperoleh dengan API melalui URL http://data.bandung.go.id/api/3/action/package_show/. Tampak pada Gambar 15 bahwa *dataset* tersebut hanya memiliki satu berkas saja sesuai dengan panjang *list* `ds_info['result']['resources']` adalah 1.

```

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
```

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

Gambar 16. URL berkas-berkas dalam dataset "jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah"

Daftar URL dari setiap berkas yang terkandung di dalam *dataset* "jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah" dapat dikumpulkan dengan menggunakan skrip berikut dan keluarannya ditampilkan pada Gambar 16.

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

Langkah kelima: Pengunduhan dataset

Pada langkah sebelumnya, URL dari berkas-berkas di dalam *dataset* yang diinginkan sudah berhasil dikumpulkan. Selanjutnya, setiap berkas tersebut diunduh pada langkah ini dengan menggunakan skrip berikut. Library *tqdm* digunakan untuk menampilkan *progress* pengunduhan data. Keluaran dari skrip ini ada pada Gambar 17. Berkas-berkas yang diunduh akan disimpan di folder aktif tempat Jupyter Notebook dijalankan. Semua berkas yang terunduh siap untuk diproses lebih lanjut oleh skrip telaah data.

```
for url in ds_urls:
    fname = url[1+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.
```

Gambar 17. Pengunduhan berkas-berkas di dalam dataset " jumlah-guru-sd-negeri-di-kota-bandung-berdasarkan-sekolah "

6.3 Pengambilan Data dengan Web Scraping

Web scraping bermakna proses ekstraksi data secara langsung dan otomatis dari suatu halaman web. Ini merupakan salah satu cara bagi ilmuwan data (*data scientist*) untuk mendapatkan data yang hanya tersedia di dalam suatu halaman web dan tidak tersedia dari sumber-sumber lain yang lebih mudah diakses. Namun demikian, ada beberapa tantangan yang akan ditemui ketika melakukan *web scraping*.

- (1) Metode *web scraping* sangat bergantung kepada struktur halaman web yang akan di-*scrape*. Namun demikian, setiap *website* memiliki struktur halamannya masing-masing yang bisa sangat berbeda satu sama lain. Jadi, penerapan *web scraping* untuk setiap situs web akan membutuhkan banyak eksperimen dan *trial-and-error*.
- (2) Konten dan struktur situs-situs web seringkali berubah secara dinamis, sehingga bisa saja terjadi bahwa sebuah skrip *web scraping* yang berjalan dengan sempurna di suatu waktu justru mengalami *error* dan kegagalan ketika dijalankan di waktu yang lain.
- (3) Konten situs web pada umumnya dibuka dalam lingkup suatu lisensi akses tertentu. Dalam hal ini, walaupun konten tersebut dapat dibaca menggunakan peramban web, tidak semua situs web mengizinkan konten web-nya di-*scrape*. Jadi, pemilik situs web bahkan bisa saja menutup akses dari alamat Internet tertentu secara sepihak ke situs webnya. Dalam kasus-kasus tertentu, pemilik situs web bahkan dapat saja menggugat pihak yang melakukan *web scraping* secara hukum.

Walaupun demikian, *web scraping* tetap merupakan salah satu cara yang dapat dipilih oleh seorang analis data untuk mendapatkan data. Pada modul ini, teknik detail untuk melakukan *web scraping* tidak dibahas. Namun, terdapat *library* Python untuk melakukan

web scraping yang dapat dieksplorasi lebih lanjut secara mandiri, yakni library `requests`² dan `beautifulsoup`.³ Tutorial daring juga dapat diikuti untuk mencoba *web scraping*, misalnya yang ada di RealPython.⁴

6.4 Pengambilan Data dari Basis Data Relasional

Sumber data alternatif yang dapat pula dipakai oleh analis data adalah basis data relasional yang tersedia secara internal di dalam organisasi. Pada prakteknya, hal ini sering dilakukan dengan bantuan *data engineer* dalam organisasi yang lebih berkompeten dalam pengelolaan data beserta infrastrukturnya di dalam organisasi. Konkritnya, seorang analis data dapat mengajukan permintaan data ke *data engineer* yang kemudian mengambilkan data tersebut dari basis data internal organisasi. Tentunya, *data engineer* akan melakukannya dengan mengeksekusi kueri SQL ke sistem basis data yang ada. Hasilnya kemudian dapat diserahkan ke analis data secara manual, misalnya berupa satu atau beberapa berkas *spreadsheet* CSV.

Namun demikian, seorang analis data dapat pula melakukannya sendiri jika memiliki akses langsung ke basis data yang bersangkutan. Detil teknisnya berada di luar lingkup modul ini, namun pada prinsipnya, selain Pandas, seorang analis data dapat menggunakan *library* Python bernama SQLAlchemy,⁵ atau alternatifnya, menggunakan *library* penghubung khusus untuk suatu mesin basis data relasional tertentu. Ada beberapa tutorial daring yang dapat diikuti jika teknik ini akan dieksplorasi lebih lanjut, misalnya dari SQLShack.com⁶ atau Medium.com.⁷

7. Telaah Data

7.1 Mengimpor Data ke Pandas dari Berkas

Pada bagian ini, data yang ada di repositori data Kaggle akan diproses menggunakan Pandas. Sebagai contoh, *dataset* “Goal Dataset – Top 5 European Leagues” akan diproses.

Pertama, jalankan dulu Jupyter Notebook di lokasi berkas CSV *dataset* tersebut. Secara otomatis, akan ada satu jendela peramban yang terbuka yang menampilkan isi folder

² <https://docs.python-requests.org/en/master/>

³ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

⁴ <https://realpython.com/beautiful-soup-web-scraper-python/>

⁵ <https://www.sqlalchemy.org>

⁶ <https://www.sqlshack.com/exploring-databases-in-python-using-pandas/>

⁷ <https://medium.com/analytics-vidhya/importing-data-from-a-mysql-database-into-pandas-data-frame-a06e392d27d7>

tempat berkas CSV *dataset* berada. Selanjutnya buka *notebook* baru dengan mengklik tombol “New” di sisi kanan dan memilih versi Python (di sini Python 3) untuk notebook baru. Beri nama notebook dengan nama yang pilih dengan cara mengklik judul notebook di sebelah kiri atas yang nilai default-nya adalah Untitled.

Pada titik ini, proses pengambilan data menggunakan Pandas dapat dimulai. Diasumsikan Anda sudah menginstal library Pandas. Jika belum, silakan rujuk kembali modul yang membahas *Data Science Tools* untuk melakukan instalasi Pandas. Pada langkah-langkah selanjutnya, akan ditampilkan potongan kode beserta keluarannya jika ada.

Impor pandas dan langsung muat isi berkas CSV yang akan dianalisis ke dalam Pandas Dataframe menggunakan perintah `read_csv()` seperti dicontohkan di bawah. Kali ini, data yang dianalisis adalah “epl-goalScorer(20-21).csv”. Hasilnya tampak seperti pada Gambar 18. Pada titik ini, data sudah berhasil dimuat ke dalam Pandas DataFrame.

```
import pandas as pd
path = "epl-goalScorer(20-21).csv"
df = pd.read_csv(path)
df
```

Unnamed: 0	id	player_name	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	position	team_title	np	
0	0	647	Harry Kane	35	3097	23	22.174859	14	7.577094	138	49	1	0	F	Tottenham	1
1	1	1250	Mohamed Salah	37	3085	22	20.250847	5	6.528526	126	55	0	0	F M S	Liverpool	1
2	2	1228	Bruno Fernandes	37	3117	18	16.019454	12	11.474996	121	95	6	0	M S	Manchester United	
3	3	453	Son Heung-Min	37	3139	17	11.023287	10	9.512992	68	75	0	0	F M S	Tottenham	1
4	4	822	Patrick Bamford	38	3085	17	18.401863	7	3.782247	107	30	3	0	F S	Leeds	1
...
517	517	9415	Jaden Philogene-Bidace	1	1	0	0.000000	0	0.000000	0	0	0	0	S	Aston Villa	
518	518	9423	Gaetano Berardi	2	113	0	0.074761	0	0.000000	1	0	0	0	D S	Leeds	
519	519	9524	Anthony Elanga	1	67	0	0.000000	0	0.000000	0	0	0	0	M	Manchester United	
520	520	9540	Femi Seriki	1	1	0	0.000000	0	0.000000	0	0	0	0	S	Sheffield United	
521	521	9552	Tyrese Francois	1	13	0	0.000000	0	0.000000	0	0	0	0	S	Fulham	

522 rows × 19 columns

Gambar 18. Keluaran perintah `read_csv` untuk dataset `epl-goalScorer(20-21).csv`

7.2 Eksplorasi Data Dasar

Eksplorasi data dasar biasanya dimulai dengan menginspeksi kolom-kolom beserta beberapa baris awal dari data. Untuk **melihat beberapa baris awal dari data**, fungsi `head()` dan `tail()` dari `DataFrame` dapat membantu. Untuk *dataset* “`epl-goalScorer(20-21).csv`”, atau mudahnya, data EPL Goal Scorer, skrip sebelumnya dapat diteruskan yang mana `DataFrame` yang menyimpan data tersebut ada pada variabel `df`, yakni

```
df.head()
```

yang hasilnya ada pada Gambar .. serta

```
df.tail()
```

yang hasilnya ada pada Gambar.

Unnamed: 0	id	player_name	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	position	team_title	npg	
0	0	647	Harry Kane	35	3097	23	22.174859	14	7.577094	138	49	1	0	F	Tottenham	19
1	1	1250	Mohamed Salah	37	3085	22	20.250847	5	6.528526	126	55	0	0	F M S	Liverpool	16
2	2	1228	Bruno Fernandes	37	3117	18	16.019454	12	11.474996	121	95	6	0	M S	Manchester United	9
3	3	453	Son Heung-Min	37	3139	17	11.023287	10	9.512992	68	75	0	0	F M S	Tottenham	16
4	4	822	Patrick Bamford	38	3085	17	18.401863	7	3.782247	107	30	3	0	F S	Leeds	15

Gambar 19. Keluaran `DataFrame` method `head()` pada data EPL Goal Scorer

Unnamed: 0	id	player_name	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	position	team_title	npg	n
517	517	9415	Jaden Philogene-Bidace	1	1	0	0.000000	0	0.0	0	0	0	S	Aston Villa	0	0.000
518	518	9423	Gaetano Berardi	2	113	0	0.074761	0	0.0	1	0	0	D S	Leeds	0	0.074
519	519	9524	Anthony Elanga	1	67	0	0.000000	0	0.0	0	0	0	M	Manchester United	0	0.000
520	520	9540	Femi Seriki	1	1	0	0.000000	0	0.0	0	0	0	S	Sheffield United	0	0.000
521	521	9552	Tyrese Francois	1	13	0	0.000000	0	0.0	0	0	0	S	Fulham	0	0.000

Gambar 20. Keluaran `DataFrame` method `tail()` pada data EPL Goal Scorer

Dengan menginspeksi keluaran pada Gambar 18, Gambar 19, dan Gambar 20, beberapa karakteristik data berikut dapat diketahui.

(1) Data terdiri dari 522 baris (indeks `DataFrame` dari 0 hingga 521).

- (2) Terdapat 19 kolom atau atribut di dalam data.
- (3) Kolom ke-0 data (indeks kolom dimulai dari 0) adalah kolom takbernama yang hanya berisi nomor urut dari 0 hingga 521.
- (4) Kolom kesatu data adalah kolom “id” yang berisi nomor *identifier* dari pemain yang dideskripsikan pada baris yang bersangkutan.

Perhatikan bahwa data dimuat Pandas seperti aslinya sesuai urutan kemunculannya dalam berkas CSV. Dalam hal ini, urutan data berdasarkan kolom ke-0.

Hal yang dapat dilakukan berikutnya adalah **menginspeksi tipe data dari masing-masing kolom**. Atribut `dtypes` dari `DataFrame` menyimpan sebuah Pandas Series yang berisi daftar seluruh kolom di dalam `DataFrame` yang bersangkutan. Perintah berikut menghasilkan tipe dari `df.dtypes`, panjang `df.dtypes` (yakni jumlah kolom dari `DataFrame df`), serta isi `df.dtypes` itu sendiri.

```
type(df.dtypes), len(df.dtypes), df.dtypes
```

Sebagaimana terlihat pada, dapat dilihat bahwa `df.dtypes` adalah sebuah Pandas Series, lalu `df` memiliki 19 kolom, serta `df` memiliki beberapa kolom numerik, yakni `int64` (bilangan bulat dengan format 64-bit) dan `float64` (bilangan *floating point* 64-bit), serta kolom non-numerik yang secara default dianggap memiliki tipe `object`. Lebih jauh lagi, dengan melihat isi kolom non-numerik, dapat diduga bahwa isi kolom-kolom tersebut sebenarnya bertipe string.

```
(pandas.core.series.Series,
19,
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)
```

Gambar 21. Tipe objek dari atribut dtypes DataFrame, panjangnya untuk kasus data EPL Goal Scorer, serta rincian isinya.

Dengan memperhatikan juga keluaran yang ditampilkan pada Gambar 19 dan Gambar 20, dapat disimpulkan bahwa atribut-atribut pada data memiliki tipe-tipe data elementer sebagai berikut.

- (1) Kolom `player_name`, `position`, dan `team_title` memiliki tipe data nominal atau kategorikal. Dengan memanfaatkan pengetahuan dari ranah sepak bola, seorang analis data bahkan dapat menyimpulkan bahwa `team_title` berisi nama-nama klub sepakbola di English Premier League, lalu `position` berisi posisi pemain saat bermain sepanjang musim, misalnya F (*forward*), M (*midfielder*), D (*defender*), dan S (*substitute*)
- (2) Kolom ke-0 yang tak bernama dan kolom `id` memiliki tipe data ordinal
- (3) Semua kolom sisanya memiliki tipe data rasio.

Di samping itu, kolom ke-0 dan kolom `id` pada dasarnya tidak memberikan informasi yang bermakna karena data-data di dalamnya sebenarnya dapat disematkan secara sembarang ke baris-baris dalam data. Dalam alam kasus ini, kebetulan saja bahwa misalnya, pemain bernama “Harry Kane” memiliki `id` 647. Oleh karena itu, analisis data selanjutnya dapat dilakukan dengan membatasi data pada semua kolom, kecuali kolom ke-0 dan kolom `id`. Hal ini dapat direalisasikan dengan membuat DataFrame baru yang

isinya seperti DataFrame df, namun tanpa kedua kolom tersebut, yakni dengan menggunakan skrip berikut yang keluarannya ada pada Gambar 22.

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

	player_name	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	position	team_title	npg	npxG	xG
0	Harry Kane	35	3097	23	22.174859	14	7.577094	138	49	1	0	F	Tottenham	19	19.130183	24.9
1	Mohamed Salah	37	3085	22	20.250847	5	6.528526	126	55	0	0	F M S	Liverpool	16	15.683834	28.9
2	Bruno Fernandes	37	3117	18	16.019454	12	11.474996	121	95	6	0	M S	Manchester United	9	8.407840	26.9
3	Son Heung-Min	37	3139	17	11.023287	10	9.512992	68	75	0	0	F M S	Tottenham	16	10.262118	20.6
4	Patrick Bamford	38	3085	17	18.401863	7	3.782247	107	30	3	0	F S	Leeds	15	16.879525	23.3
...
517	Jaden Philogene-Bidace	1	1	0	0.000000	0	0.000000	0	0	0	0	S	Aston Villa	0	0.000000	0.0
518	Gaetano Berardi	2	113	0	0.074761	0	0.000000	1	0	0	0	D S	Leeds	0	0.074761	0.2
519	Anthony Elanga	1	67	0	0.000000	0	0.000000	0	0	0	0	M	Manchester United	0	0.000000	0.0
520	Femi Seriki	1	1	0	0.000000	0	0.000000	0	0	0	0	S	Sheffield United	0	0.000000	0.0
521	Tyrese Francois	1	13	0	0.000000	0	0.000000	0	0	0	0	S	Fulham	0	0.000000	0.0

522 rows × 17 columns

Gambar 22. Hasil membuat DataFrame baru dengan membuang kolom ke-0 dan kolom id dari DataFrame df.

Hal lain yang juga dapat dilakukan adalah menampilkan data dengan mengikuti urutan tertentu. Data yang ditampilkan pada Gambar 22 nampaknya terurut berdasarkan jumlah gol (kolom goals). Jika diinginkan, skrip berikut dapat dipakai untuk mengurutkan data berdasarkan nama pemain dengan urutan menaik dan ditampilkan 10 baris pertama dari hasilnya. Keluaran ada pada Gambar 23.

```
df1 = df_noid.sort_values(by="player_name",ascending=True)
df1.head(10)
```

	player_name	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	position	team_title	npg	npxG	xG
154	Aaron Connolly	16	755	2	4.412464	1	0.149897	22	5	0	0	F M S	Brighton	2	4.412464	4.1
281	Aaron Cresswell	35	3086	0	0.883464	8	7.347331	19	57	3	0	D	West Ham	0	0.883464	10.6
390	Aaron Ramsdale	37	3330	0	0.000000	0	0.053571	0	1	1	0	GK	Sheffield United	0	0.000000	2.1
139	Aaron Wan-Bissaka	34	3060	2	0.932454	4	2.547663	7	31	3	0	D	Manchester United	2	0.932454	12.1
126	Abdoulaye Doucouré	28	2409	2	2.369523	3	2.381616	18	20	6	0	M	Everton	2	2.369523	10.1
380	Aboubakar Kamara	11	303	0	0.654920	0	0.328969	4	6	1	1	F M S	Fulham	0	0.654920	1.1
163	Adam Lallana	29	1528	1	1.614306	1	2.628758	22	25	0	0	F M S	Brighton	1	1.614306	9.1
242	Adam Webster	28	2506	1	1.272957	0	0.253216	25	5	4	0	D	Brighton	1	1.272957	7.6
119	Adama Traoré	36	2604	2	1.995262	2	5.228081	41	54	4	0	D F M S	Wolverhampton Wanderers	2	1.995262	9.1
75	Ademola Lookman	34	2765	4	6.251116	4	5.258407	69	61	5	0	F M S	Fulham	4	5.489947	15.1

Gambar 23. Data terurut berdasarkan nama pemain.

Lalu, skrip berikut mengurutkan data berdasarkan jumlah *assist* (umpan terakhir sebelum terjadinya gol) dengan urutan menaik, lalu jika sama, berdasarkan nama klubnya dengan urutan menurun, lalu ditampilkan 10 baris pertama dari hasilnya seperti pada Gambar 24.

```
df1 = df_noid.sort_values(by=["assists", "team_title"], \
                           ascending=[False,True])
df1.head(10)
```

	player_name	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	position	team_title	npg	npxG	xG
0	Harry Kane	35	3097	23	22.174859	14	7.577094	138	49	1	0	F	Tottenham	19	19.130183	24.96
2	Bruno Fernandes	37	3117	18	16.019454	12	11.474996	121	95	6	0	M S	Manchester United	9	8.407840	26.91
58	Kevin De Bruyne	24	1918	5	9.908440	11	10.003763	79	72	1	0	M S	Manchester City	3	7.624933	21.07
51	Jack Grealish	26	2187	6	5.192684	10	9.334137	50	81	5	0	F M S	Aston Villa	6	5.192684	17.48
3	Son Heung-Min	37	3139	17	11.023287	10	9.512992	68	75	0	0	F M S	Tottenham	16	10.262118	20.67
57	Raphinha	30	2369	6	6.219143	9	9.524861	67	65	3	0	M S	Leeds	6	6.219143	16.78
6	Jamie Vardy	34	2848	15	19.942946	9	5.087882	82	28	1	0	F S	Leicester	7	13.092427	18.22
15	Marcus Rashford	37	2941	11	9.579710	9	4.185122	79	44	4	0	F M S	Manchester United	11	9.579710	20.44
83	Pascal Groß	33	2379	3	5.010526	8	5.368290	32	69	3	0	D M S	Brighton	0	1.965887	10.40
49	Timo Werner	35	2605	6	13.432796	8	6.667277	80	36	2	0	F M S	Chelsea	6	13.432796	20.53

Gambar 24. Data terurut berdasarkan jumlah assist (menurun) lalu berdasarkan nama klub (menaik).

7.3 Eksplorasi Data secara Visual

Eksplorasi data juga dapat dilakukan secara visual. Terdapat beberapa *library* Python yang dapat dipakai untuk membuat visualisasi data yang diinginkan. Pembahasan lebih terperinci mengenai topik ini diberikan di dalam modul terpisah, yakni Modul Visualisasi.

7.4 Deskripsi Data secara Statistik

Metode eksplorasi data yang lain adalah dengan menerapkan konsep-konsep dari ilmu statistika. Pandas menyediakan cukup banyak fungsi-fungsi statistika yang dapat diterapkan pada suatu DataFrame. Tabel 3 berisi daftar fungsi-fungsi statistika pada Pandas DataFrame. Beberapa konsep statistika yang terkait dengan fungsi-fungsi di bawah tidak dibahas secara detil di modul ini. Sebagai gantinya, diberikan tautan-tautan ke halaman Wikipedia yang memberikan bahasan pengantar yang cukup baik.

Tabel 3. Fungsi-fungsi statistika pada Pandas DataFrame

Nama fungsi	Fungsi mengembalikan...
count	banyaknya butir data yang bukan NA (NA = <i>not available</i>).
sum	jumlahan butir-butir data.
mean	rerata (aritmetik) ⁸ butir-butir data.
mad	rerata simpangan absolut (<i>mean absolute deviation</i>) ⁹ dari butir-butir data.
median	median ¹⁰ (aritmetik) dari butir-butir data.
min	nilai terkecil/minimum dari butir-butir data.
max	nilai terbesar/maksimum dari butir-butir data.
mode	nilai modus ¹¹ dari butir-butir data.
abs	nilai absolut ¹² numerik setiap butir data.
prod	hasil perkalian setiap butir data.
quantile	nilai pada kuantil ¹³ tertentu dari butir-butir data; argumen fungsi adalah kuantil yang diinginkan antara 0 hingga 1; nilai kuartil pertama = nilai kuantil pada posisi 0.25.
std	nilai simpangan baku sampel ¹⁴ (bukan populasi) menggunakan koreksi Bessel; kumpulan butir data yang dihitung simpangan bakunya dianggap sebagai kumpulan sampel, bukan seluruh populasi.

⁸ <https://en.wikipedia.org/wiki/Mean>

⁹ https://en.wikipedia.org/wiki/Average_absolute_deviation

¹⁰ <https://en.wikipedia.org/wiki/Median>

¹¹ [https://en.wikipedia.org/wiki/Mode_\(statistics\)](https://en.wikipedia.org/wiki/Mode_(statistics))

¹² https://en.wikipedia.org/wiki/Absolute_value

¹³ <https://en.wikipedia.org/wiki/Quantile>

¹⁴ https://en.wikipedia.org/wiki/Standard_deviation#Corrected_sample_standard_deviation

var	nilai varian sampel; ¹⁵ kumpulan butir data yang dihitung varian-nya dianggap sebagai kumpulan sampel.
sem	galat standar dari rerata. ¹⁶
skew	nilai ukuran kecondongan ¹⁷ (<i>skewness</i>) dari distribusi
kurt	nilai ukuran keruncingan ¹⁸ (<i>kurtosis</i>) dari distribusi
cumsum	jumlahan kumulatif data.
cumprod	perkalian kumulatif data.
cummax	nilai maksimum kumulatif data
cummin	nilai minimum kumulatif data.

7.4.1 Ringkasan Statistik dengan Fungsi `describe()`

Untuk mendapatkan ringkasan statistik secara cepat, DataFrame menyediakan fungsi `describe()` yang akan menghasilkan sebuah DataFrame baru yang berisi ringkasan statistik dari DataFrame yang padanya `describe()` diterapkan. Sebagai contoh, fungsi tersebut dapat diterapkan pada DataFrame `df_noid` yang sudah didefinisikan sebelumnya.

```
df_noid.describe()
```

Hasilnya dapat dilihat pada Gambar 25. Perhatikan bahwa fungsi `describe()` di atas hanya menghasilkan ringkasan statistik pada kolom-kolom bertipe numerik. Ini juga bersesuaian dengan besaran-besaran statistika yang ditampilkan yang sebagian besar hanya dapat dipakai pada tipe data interval dan rasio.

	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	npg	npxG
count	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000
mean	19.643678	1420.068966	1.862069	2.000806	1.289272	1.376029	17.379310	12.963602	2.061303	0.091954	1.668582	1.821450
std	11.619836	1031.604819	3.338851	3.317946	2.083350	1.886510	21.572664	16.164361	2.203661	0.295800	2.909929	2.931176
min	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	10.000000	470.250000	0.000000	0.074668	0.000000	0.049245	2.000000	1.000000	0.000000	0.000000	0.000000	0.074668
50%	21.000000	1342.000000	1.000000	0.737295	0.000000	0.691122	10.000000	7.000000	2.000000	0.000000	0.500000	0.715585
75%	30.000000	2319.000000	2.000000	2.053378	2.000000	2.050509	23.750000	19.000000	3.000000	0.000000	2.000000	1.945799
max	38.000000	3420.000000	23.000000	22.174859	14.000000	11.474996	138.000000	95.000000	12.000000	2.000000	19.000000	19.130183

Gambar 25. Keluaran fungsi `describe()`

¹⁵ https://en.wikipedia.org/wiki/Variance#Unbiased_sample_variance

¹⁶ https://en.wikipedia.org/wiki/Standard_error#Standard_error_of_the_mean

¹⁷ <https://en.wikipedia.org/wiki/Skewness>

¹⁸ <https://en.wikipedia.org/wiki/Kurtosis>

Untuk mendapatkan juga ringkasan statistik pada kolom-kolom non-numerik, parameter `include='all'` dapat diberikan pada fungsi `describe()`, yakni

```
df_noid.describe(include='all')
```

yang menghasilkan keluaran pada Gambar 26 dan Gambar 27.

	player_name	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	position	t
count	522	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522	
unique	522	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	14	
top	Willian José	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	M S	
freq	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	106	
mean	NaN	19.643678	1420.068966	1.862069	2.000806	1.289272	1.376029	17.379310	12.963602	2.061303	0.091954	NaN	
std	NaN	11.619836	1031.604819	3.338851	3.317946	2.083350	1.886510	21.572664	16.164361	2.203661	0.295800	NaN	
min	NaN	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	
25%	NaN	10.000000	470.250000	0.000000	0.074668	0.000000	0.049245	2.000000	1.000000	0.000000	0.000000	NaN	
50%	NaN	21.000000	1342.000000	1.000000	0.737295	0.000000	0.691122	10.000000	7.000000	2.000000	0.000000	NaN	
75%	NaN	30.000000	2319.000000	2.000000	2.053378	2.000000	2.050509	23.750000	19.000000	3.000000	0.000000	NaN	
max	NaN	38.000000	3420.000000	23.000000	22.174859	14.000000	11.474996	138.000000	95.000000	12.000000	2.000000	NaN	

Gambar 26. Keluaran fungsi `describe()` dengan parameter `include='all'`; untuk melihat besaran statistik semua kolom, keluaran tersebut dapat digeser (*di-scroll*) ke kanan di dalam Jupyter Notebook.

Is	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	position	team_title	npg	npxG	xGChain	xGBuildup
10	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522.000000	522	522	522.000000	522.000000	522.000000	522.000000
N	NaN	NaN	NaN	NaN	NaN	NaN	NaN	14	28	NaN	NaN	NaN	NaN
N	NaN	NaN	NaN	NaN	NaN	NaN	NaN	M S	West Bromwich Albion	NaN	NaN	NaN	NaN
N	NaN	NaN	NaN	NaN	NaN	NaN	NaN	106	28	NaN	NaN	NaN	NaN
19	2.000806	1.289272	1.376029	17.379310	12.963602	2.061303	0.091954	NaN	NaN	1.668582	1.821450	5.663368	3.455060
11	3.317946	2.083350	1.886510	21.572664	16.164361	2.203661	0.295800	NaN	NaN	2.909929	2.931176	5.600249	3.376584
10	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	NaN	0.000000	0.000000	0.000000	0.000000
10	0.074668	0.000000	0.049245	2.000000	1.000000	0.000000	0.000000	NaN	NaN	0.000000	0.074668	1.191391	0.720353
10	0.737295	0.000000	0.691122	10.000000	7.000000	2.000000	0.000000	NaN	NaN	0.500000	0.715585	4.252738	2.656397
10	2.053378	2.000000	2.050509	23.750000	19.000000	3.000000	0.000000	NaN	NaN	2.000000	1.945799	8.308002	5.254647
10	22.174859	14.000000	11.474996	138.000000	95.000000	12.000000	2.000000	NaN	NaN	19.000000	19.130183	28.968234	18.323006

Gambar 27. Keluaran perintah `describe(include='all')` yang sama dengan Gambar 26 setelah digeser ke kanan.

Setiap baris keluaran `describe(include='all')` pada Gambar 26 dan Gambar 27 memiliki makna sebagai berikut yang mana besaran **count** berlaku untuk semua kolom, besaran **unique**, **top**, dan **count** berlaku hanya pada kolom-kolom non-numerik (seperti *string* atau *timestamps*), sementara besaran-besaran lainnya berlaku hanya pada kolom-kolom numerik. Di samping itu, `describe(include='all')` dapat juga mencakup besaran **first** dan **last** yang hanya berlaku untuk kolom-kolom yang berisi butir-butir data *timestamps* (atau titik-titik waktu). Besaran **first** dan **last** tersebut menyatakan titik waktu awal dan akhir dari suatu *time series*.

- (1) Baris **count** menyatakan banyaknya butir data pada kolom yang bersangkutan, yakni sama dengan jumlah baris dalam *dataset*. Besaran **count** berlaku untuk semua kolom (numerik dan non-numerik).
- (2) Baris **unique** menyatakan banyaknya butir data unik pada kolom yang bersangkutan, yakni seperti **count** namun setelah butir data duplikat dibuang. Besaran **unique** hanya memiliki makna pada kolom non-numerik; besaran ini bernilai NaN (*not a number*) pada kolom numerik.
- (3) Baris **top** menyatakan butir data yang paling sering muncul pada kolom yang bersangkutan, yakni modus dari kolom tersebut. Jika kolom tersebut ternyata memiliki lebih dari satu modus, maka salah satu akan dipilih secara acak oleh Pandas untuk dikembalikan sebagai keluaran di sini. Besaran ini hanya memiliki makna pada kolom non-numerik.
- (4) Baris **freq** menyatakan frekuensi kemunculan dari modus pada kolom yang bersangkutan. Besaran ini hanya memiliki makna pada kolom non-numerik.
- (5) Baris **mean** menyatakan rerata aritmetik dari butir-butir data pada kolom yang bersangkutan.
- (6) Baris **std** menyatakan simpangan baku sampel (yakni dengan koreksi Bessel) dari butir-butir data pada kolom yang bersangkutan.
- (7) Baris **min** menyatakan butir data terkecil dari butir-butir data pada kolom yang bersangkutan.
- (8) Baris **25%** menyatakan kuartil pertama dari butir-butir data pada kolom yang bersangkutan.
- (9) Baris **50%** menyatakan kuartil kedua atau median dari butir-butir data pada kolom yang bersangkutan.
- (10) Baris **75%** menyatakan kuartil ketiga dari butir-butir data pada kolom yang bersangkutan.
- (11) Baris **max** menyatakan butir data terbesar dari butir-butir data pada kolom yang bersangkutan.

Dari Gambar 26 dan Gambar 27, dapat diketahui beberapa hal berikut yang tersirat dari data.

- (1) Terdapat 522 pemain yang dideskripsikan di dalam data karena ada 522 baris (**count**) dan ada 522 nilai unik (**unique**) pada kolom `player_name`.

- (2) Terdapat 28 nama tim/klub menurut kolom `team_title`. Jika digabungkan dengan *background knowledge* terkait English Premier League yang hanya beranggotakan 20 klub pada periode liga saat data tersebut diambil, maka terdapat ketidakcocokan data dengan keadaan sebenarnya. Hal ini dapat diselidiki lebih lanjut dengan menginspeksi isi kolom `team_title`.
- (3) Rerata (**mean**) gol per pemain adalah sekitar 1.86, sedangkan di dalam data, ada pemain yang tidak mencetak gol sama sekali, yakni dengan nilai kolom `goals` 0, serta pemain dengan jumlah gol terbanyak mencetak 23 gol sepanjang musim. Hal ini menunjukkan bahwa distribusi jumlah gol per pemain condong (*skew*) ke arah “kiri”, yakni mayoritas pemain hanya mencetak sedikit gol atau bahkan tidak sama sekali. Hanya sedikit pemain yang mencetak 6 gol atau lebih.
- (4) Beberapa kolom memiliki nilai rerata (**mean**) dan simpangan baku (**std**) sedemikian hingga nilai selisih rerata dengan simpangan baku, yakni **mean - std** yang negatif. Padahal, kolom-kolom tersebut mengandung data-data numerik dengan tipe rasio yang mana nilai nol sejatinya atau nilai terendah yang mungkin adalah 0. Artinya, banyak sekali pemain memiliki nilai rendah atau bahkan nol pada kolom-kolom tersebut.
- (5) Terdapat sangat sedikit pemain yang pernah mendapatkan kartu merah sepanjang musim. Hal ini terlihat dari rerata jumlah kartu merah yang berada di sekitar 0.09 sedangkan pemain dengan kartu merah terbanyak mendapatkan 2 kartu merah.

Hal-hal di atas merupakan contoh beberapa observasi yang dapat dilakukan pada data yang dianalisis. Seorang analis data tentunya dapat saja menggali beberapa observasi lain dari data. Selanjutnya, konsep-konsep yang disebutkan di atas diperkenalkan secara lebih formal.

7.4.2 Deskripsi Pusat Data: Rerata Aritmetik, Median, dan Modus

Butir-butir data setiap kolom dapat dipandang sebagai sampel dari suatu distribusi statistik tertentu. Deskripsi pusat data pada dasarnya memberikan gambaran mengenai lokasi tempat berkumpulnya kebanyakan butir data pada distribusi tersebut. Terdapat tiga besaran pusat data yang paling banyak dipergunakan, yakni rerata aritmetik (*mean*), median, dan modus. Walaupun modus dapat diterapkan pada data-data numerik, sebagaimana terlihat pada Gambar 26 dan Gambar 27, Pandas mengasumsikan

penggunaan konsep modus hanya pada data-data non-numerik. Tidak hanya itu, konsep rerata aritmetik dan median hanya diterapkan pada data-data numerik.

Rerata Aritmetik

Rerata aritmetik (*mean*) dari sekumpulan bilangan adalah hasil pembagian antara jumlah dari semua bilangan tersebut dengan banyaknya bilangan dalam kumpulan tersebut. Andaikan $S = \{x_1, \dots, x_N\}$ adalah kumpulan N buah bilangan (mungkin dengan duplikat), maka rerata aritmetik dari S , yang dinyatakan dengan notasi μ_S atau \bar{x} , didefinisikan secara matematis sebagai:

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

Jadi, untuk kumpulan bilangan {9,36,30,45,50,28}, rerata aritmetiknya adalah

$$\frac{9 + 36 + 30 + 45 + 50 + 28}{6} = \frac{198}{6} = 33$$

Dengan menggunakan Pandas, rerata aritmetik dapat dihitung dengan fungsi `mean()`. Misalnya untuk menghitung rerata gol yang dicetak seorang pemain, pertandingan yang melibatkan pemain yang bersangkutan, banyaknya *assist* setiap pemain, serta banyaknya tembakan ke gawang lawan yang dilakukan setiap pemain, skrip berikut dapat digunakan.

```
df_noid[['goals', 'games', 'assists', 'shots']].mean()
```

Hasilnya ada pada Gambar 28

```
goals      1.862069
games      19.643678
assists     1.289272
shots      17.379310
dtype: float64
```

Gambar 28. Keluaran perintah `mean()`

Rerata aritmetik di atas merupakan formulasi yang lazim digunakan oleh banyak orang. Dalam sains data, konsep rerata aritmetik dapat dipergunakan untuk butir-butir data bertipe interval dan rasio. Rerata aritmetik merupakan ukuran pusat data karena nilainya berada di “tengah” kumpulan data. Hal ini terlihat dari sifat rerata aritmetik,

yakni total jarak setiap bilangan dalam kumpulan ke rerata aritmetikanya selalu sama dengan 0, atau

$$\sum_i (\mu_S - x_i) = 0$$

Nilai rerata aritmetik suatu kumpulan bilangan dapat dipakai sebagai wakil atau representasi yang baik dari kumpulan bilangan tersebut apabila distribusi bilangannya tidak bersifat asimetris atau *skew*.

Median

Median dari suatu kumpulan data adalah butir data yang posisinya berada di tengah-tengah. Lebih tepatnya, jika $S = \{x_1, \dots, x_N\}$ adalah kumpulan N buah butir data, maka median dari S adalah butir data x_j sedemikian hingga separuh dari butir-butir data dari S memiliki nilai yang lebih kecil atau sama dengan x_j . Jadi, terdapat paling banyak $N/2$ butir data x_i sehingga $x_i \leq x_j$, dan terdapat paling banyak $N/2$ butir data x_k sehingga $x_k \geq x_j$. Formulasi ini dapat dipecah kasusnya tergantung apakah N ganjil atau genap.

- (1) Jika N ganjil, maka median kumpulan data adalah butir data pada posisi atau *ranking* ke- $\left(\frac{N+1}{2}\right)$, setelah butir-butir data tersebut diurutkan. Contohnya, median dari kumpulan data $\{9, 28, 30, 31, 36, 45, 50\}$ adalah 31.
- (2) Jika N genap, maka median kumpulan data adalah rerata aritmetik antara butir data pada posisi ke- $\left(\frac{N}{2}\right)$ dan butir data pada posisi ke- $\left(\frac{N}{2} + 1\right)$ setelah butir-butir data tersebut diurutkan. Contohnya, median dari kumpulan data $\{9, 28, 30, 36, 45, 50\}$ adalah rerata antara 30 dan 36, yakni 33.

Definisi di atas menunjukkan bahwa median dapat dipakai untuk data yang bertipe ordinal, interval, dan rasio, namun tidak untuk data yang bertipe nominal atau kategorikal. Hal ini disebabkan oleh tiadanya konsep urutan pada tipe data nominal. Perhatikan pula bahwa data bertipe ordinal tidak harus berupa bilangan, karena yang penting kumpulan datanya mengandung urutan secara implisit atau eksplisit. Namun demikian, Pandas secara *default* mengasumsikan bahwa penghitungan median dilakukan hanya pada data numerik. Misalnya, untuk menghitung median dari gol, pertandingan, *assist* dan jumlah tembakan (*shots*) setiap pemain, skrip berikut dapat digunakan.

```
df_noid[['goals', 'games', 'assists', 'shots']].median()
```

Hasilnya ada pada Gambar 29.

```
goals      1.0  
games     21.0  
assists    0.0  
shots     10.0  
dtype: float64
```

Gambar 29. Keluaran perintah median()

Median merupakan ukuran pusat data yang bersifat *robust*, artinya tidak dipengaruhi oleh adanya pencilan (*outliers*): jika ada pencilan, nilai median tidak banyak bergeser ke arah “kiri” atau “kanan” distribusi. Ini berbeda dengan rerata aritmetik yang tidak bersifat *robust*. Oleh sebab itu, median lebih cocok dipakai sebagai wakil dari distribusi data daripada rerata aritmetik apabila distribusi tersebut bersifat condong (*skew*) ke satu arah tertentu. Misalnya, pada data tingkat pendapatan penduduk Indonesia, distribusi yang diperoleh cenderung condong ke satu arah karena mayoritas penduduk Indonesia berpendapatan menengah ke rendah. Di sini, nilai pendapatan median (*median income*) lebih sesuai untuk menggambarkan pusat data daripada nilai pendapatan rata-rata.

Hal serupa dapat terlihat juga ketika membandingkan Gambar 28 dan Gambar 29. Median dari jumlah *assist* ternyata 0 (yang juga merupakan nilai minimum) sementara reratanya 1.29. Hal ini menunjukkan bahwa distribusi jumlah *assist* per pemain sangat condong ke kiri. Sifat ini juga muncul pada data jumlah gol, serta jumlah tembakan ke gawang lawan. Sementara itu, distribusi jumlah pertandingan cenderung ke kanan karena nilai median-nya berada di sebelah kanan nilai reratanya.

Modus

Modus merupakan ukuran pusat data yang lazim dipakai untuk data-data bertipe nominal atau kategorikal. Secara formal, modus dari suatu kumpulan data adalah butir data yang paling sering muncul di dalam kumpulan data tersebut. Dengan kata lain, modus adalah butir data yang memiliki frekuensi kemunculan tertinggi. Tentunya, suatu kumpulan data dapat saja memiliki lebih dari satu modus, dan dalam hal ini kumpulan data tersebut disebut sebagai kumpulan data *multimodal*.

Pada Pandas, modus secara *default* dipakai pada data non-numerik yang biasanya dapat digolongkan sebagai data nominal. Walaupun demikian, sebenarnya konsep modus

tetap dapat diterapkan pada data numerik yang distribusinya diskrit. Contohnya, himpunan data {1,2,2,3,4,4,7,8} memiliki modus 2 dan 4. Contoh yang lain, pada data EPL Goal Scorer yang dipakai sebelumnya, nilai modus pada kolom `team_title` menunjukkan tim yang memiliki paling banyak pemain yang muncul pada data.

```
df_noid[['team_title']].mode()
```

Hasilnya ada pada Gambar 30 yang menunjukkan bahwa ada dua tim dengan jumlah pemain terbanyak, yakni Everton dan West Bromwich Albion.

team_title	
0	Everton
1	West Bromwich Albion

Gambar 30. Keluaran fungsi `mode()`

Untuk data numerik dengan distribusi kontinu, definisi modus standar tersebut perlu disesuaikan. Misalnya, penentuan modus pada kumpulan data {0.935, ..., 1.134, ..., 2.643, ..., 3.459, ..., 3.995, ... }, tidak dapat dilakukan dengan mencari nilai paling sering muncul. Pada kasus ini, terdapat dua solusi. Pertama, kumpulan data didiskretisasi terlebih dahulu sehingga dapat diperlakukan sebagai data nominal. Teknik ini dibahas pada modul Persiapan Data (*Data Preparation*). Kedua, pada kumpulan data, dilakukan *kernel density estimation*¹⁹ yang merupakan teknik yang lebih rumit dan berada di luar cakupan pembahasan modul ini. Untuk distribusi data yang tepat simetrik, seperti distribusi normal, nilai modus akan sama dengan nilai rerata aritmetik dan nilai median.

7.4.3 Deskripsi Sebaran Data: Rentang, Kuartil, Simpangan Baku, dan Pencilan

Jika deskripsi pusat data menunjukkan lokasi butir-butir data secara umum berkumpul, maka deskripsi sebaran data menggambarkan seberapa jauh butir-butir data menyebar dari pusat data. Ada beberapa besaran yang dapat digunakan untuk memberikan gambaran tentang sebaran data. Besaran-besaran tersebut antara lain meliputi rentang, kuantil, simpangan baku, varian, dan pencilan.

¹⁹ https://en.wikipedia.org/wiki/Kernel_density_estimation

Rentang

Rentang (*range*) atau jangkauan didefinisikan sebagai selisih antara nilai maksimum dan minimum pada kumpulan data. Nilai rentang yang besar dapat menggambarkan bahwa data cenderung tersebar, dan sebaliknya rentang yang kecil dapat menunjukkan bahwa data cenderung mengumpul. Namun demikian, ini tidak sepenuhnya dapat dijadikan pegangan, khususnya jika nilai maksimum atau minimum data ternyata merupakan pencilan. Tidak ada fungsi khusus dalam Pandas untuk menghitung rentang karena ini dengan mudah dapat dihitung memakai fungsi `min()` dan `max()`. Di sisi lain, karena nilai rentang hanya bergantung pada dua butir data saja, maka besaran ini biasanya hanya cocok dipakai untuk *dataset* berukuran kecil.

Kuantil, kuartil, persentil

Sebuah kuantil (*quantile*) dari suatu kumpulan data didefinisikan sebagai sebuah nilai titik potong yang menentukan berapa banyak butir data yang bernilai lebih kecil darinya dan berapa banyak yang lebih besar darinya. Untuk setiap bilangan bulat $k \geq 2$, k -kuantil adalah nilai-nilai yang besarnya membagi himpunan data menjadi k bagian yang berukuran sama. Untuk setiap k , terdapat $k - 1$ nilai atau butir data yang berfungsi sebagai k -kuantil. Ada beberapa istilah khusus untuk kuantil.

- (1) 2-kuantil hanya terdiri dari satu butir data yakni median yang membagi kumpulan data menjadi dua bagian yang sama besar: separuh berada di bawahnya dan separuh sisanya berada di atasnya.
- (2) 4-kuantil terdiri dari tiga titik atau tiga butir data yang disebut kuartil. Ketiga kuartil tersebut secara bersama-sama membagi kumpulan data menjadi empat bagian yang sama besar. Kuartil pertama membagi data sehingga 25% data berada di bawahnya dan 75% data berada di atasnya. Kuartil kedua membagi data sehingga 50% data berada di bawahnya dan 50% berada di atasnya. Lalu, kuartil ketiga membagi data sehingga 75% data berada di bawahnya dan 25% data berada di atasnya.
- (3) 100-kuantil, disebut juga persentil, meliputi 99 nilai yang secara bersama-sama membagi data menjadi 100 bagian yang sama besar.

Pandas menyediakan fungsi `quantile()` yang meng-generalisasi konsep di atas. Salah satu parameter pentingnya adalah `q` yang bernilai di antara 0 dan 1. Contohnya,

untuk mendapatkan kuartil ketiga pada sebaran jumlah gol yang dicetak pemain, maka perintahnya adalah sesuai skrip berikut.

```
df_noid[['goals']].quantile(q=0.75)
```

Hasilnya, yakni pada Gambar 31, menunjukkan bahwa 75% data pemain mencetak tidak lebih dari 2 gol.

```
goals      2.0  
Name: 0.75, dtype: float64
```

Gambar 31. Keluaran fungsi quantile()

Simpangan Baku dan Varian

Ukuran sebaran data lain yang lazim dipakai adalah simpangan baku dan varian. Varian didefinisikan sebagai rerata dari jumlah kuadrat jarak antara setiap butir data dengan rerata kumpulan data. Tepatnya, jika diberikan kumpulan data $S = \{x_1, \dots, x_N\}$ berisi N butir data dengan rerata μ_S , maka varian dari S , dinotasikan dengan σ_S^2 , didefinisikan sebagai berikut.

$$\sigma_S^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_S)^2$$

Pembagi $N - 1$ pada formula di atas disebut sebagai faktor koreksi Bessel yang digunakan untuk butir-butir data yang merupakan sampel dari populasi sesungguhnya yang tidak diketahui pasti distribusinya.

Simpangan baku dari himpunan S dinotasikan dengan σ_S dan didefinisikan sebagai akar kuadrat dari varian dari S , yakni

$$\sigma_S = \sqrt{\sigma_S^2}$$

Nilai simpangan baku yang besar menunjukkan bahwa data tersebar jauh dari rerata. Di samping itu, nilai simpangan baku dapat dipandang sebagai derajat ketidakpastian pengukuran data. Contohnya, jika nilai suatu besaran diukur secara berulang menggunakan instrumen pengukuran yang sama, maka nilai-nilai yang didapat bisa jadi bervariasi. Di sini, nilai yang biasanya diambil sebagai butir data adalah rerata nilai tersebut. Kemudian, simpangan baku dari nilai-nilai tersebut menunjukkan seberapa

presisi pengukuran yang dilakukan: jika simpangan bakunya besar, maka presisi pengukuran rendah.

Pandas menyediakan fungsi `var()` dan `std()` untuk menghitung varian dan simpangan baku. Contohnya, varian dan simpangan baku dari jumlah gol yang dicetak setiap pemain diperoleh dengan skrip berikut dan hasilnya ada pada Gambar 32

```
df_noid[['goals']].var(), df_noid[['goals']].std()
```

```
(goals      11.147925  
dtype: float64,  
goals       3.338851  
dtype: float64)
```

Gambar 32. Keluaran fungsi `var()` dan `std()`

Pencilan

Pencilan (*outlier*) merupakan butir data yang sangat berbeda dengan kebanyakan butir data lainnya di dalam kumpulan. Pencilan dapat muncul karena kesalahan yang terjadi saat pengambilan data di lapangan atau kerusakan alat pengukuran. Jika ini yang terjadi, analisis data biasanya dapat membuang pencilan dari data sebelum diproses lebih lanjut. Namun, pencilan juga dapat muncul bukan karena adanya kesalahan pengukuran, tetapi karena memang benar terkandung di dalam datanya. Hal ini dapat mengindikasikan adanya anomali yang menarik untuk dianalisis lebih lanjut.

Tidak ada kriteria yang pasti apakah suatu butir data dapat digolongkan sebagai sebuah pencilan. Oleh sebab itu, dalam sains data, terdapat beberapa metode atau kriteria yang menjadi alternatif untuk mendeteksi anomali. Beberapa di antaranya adalah *3-sigma rule*, *Tukey's fences*, visualisasi, uji Grubb, uji Dixon Q, algoritma Expectation Maximization, jarak k-nearest neighbor, *local outlier factor* berbasis *clustering* dan lain-lain. Di sini kita akan membahas dua yang pertama yang cukup mudah diimplementasikan. Teknik visualisasi untuk mendeteksi pencilan akan dibahas secara tersendiri di modul Visualisasi, sedangkan teknik-teknik lainnya berada di luar lingkup modul ini.

Deteksi pencilan menggunakan *3-sigma rule*²⁰ terdiri dari kriteria pencilan berikut. Andaikan S adalah kumpulan nilai yang akan dicari pencilannya, μ_S adalah rerata aritmetik dari S , serta σ_S adalah simpangan baku. Jika S mengikuti distribusi yang mendekati **normal**/*Gaussian*, maka ada tiga sifat yang berlaku yang mana sifat kedua atau ketiga dapat dipakai sebagai kriteria penentu pencilan.

- (1) Untuk suatu butir data $x_i \in S$, peluang x_i untuk berada dalam interval $\mu_S - \sigma_S \leq x_i \leq \mu_S + \sigma_S$ adalah sekitar 68.27%.
- (2) Untuk suatu butir data $x_i \in S$, peluang x_i untuk berada dalam interval $\mu_S - 2\sigma_S \leq x_i \leq \mu_S + 2\sigma_S$ adalah sekitar 95.45%; jadi jika $x_i < \mu_S - 2\sigma_S$ atau $x_i > \mu_S + 2\sigma_S$, maka x_i adalah pencilan. Dengan kata lain, x_i adalah pencilan jika butir data serupa x_i hanya muncul 1 kali dari 22 kali observasi.
- (3) Untuk suatu butir data $x_i \in S$, peluang x_i untuk berada dalam interval $\mu_S - 3\sigma_S \leq x_i \leq \mu_S + 3\sigma_S$ adalah sekitar 99.73%; jadi jika $x_i < \mu_S - 3\sigma_S$ atau $x_i > \mu_S + 3\sigma_S$, maka x_i adalah pencilan. Dengan kata lain, x_i adalah pencilan jika butir data serupa x_i hanya muncul 1 kali dari 370 kali observasi.

Berikut contoh skrip yang mengimplementasikan kriteria ketiga *3-sigma rule* di atas pada data jumlah gol di *dataset* EPL Goal Scorer. (Catatan: distribusi jumlah gol sebenarnya tidak mengikuti distribusi normal; skrip di bawah hanya untuk mengilustrasikan implementasi kriteria ketiga *3-sigma rule*). Hasil dapat dilihat pada Gambar 33.

```
mean = df_noid[['goals']].mean()
stdev = df_noid[['goals']].std()
iso = (df_noid[['goals']] < mean - 3*stdev) \
      | (df_noid[['goals']] > mean + 3*stdev)

df1 = df_noid[['player_name', 'goals']].assign(is_outlier=iso)
df1.loc[df1['is_outlier']]
```

Kriteria *3-sigma rule* di atas memiliki beberapa kekurangan. Pertama, ia mengasumsikan data berasal dari suatu distribusi normal (yang belum tentu benar).

²⁰ https://en.wikipedia.org/wiki/68-95-99.7_rule

Kedua, ia bergantung pada nilai rerata dan simpangan baku sampel, dan kedua nilai ini tidak *robust* karena pencilan justru dapat membuat nilai keduanya bergeser. Ketiga, jika ukuran sampel kecil, deteksi pencilan tidak dapat dilakukan.

	player_name	goals	is_outlier
0	Harry Kane	23	True
1	Mohamed Salah	22	True
2	Bruno Fernandes	18	True
3	Son Heung-Min	17	True
4	Patrick Bamford	17	True
5	Dominic Calvert-Lewin	16	True
6	Jamie Vardy	15	True
7	Ollie Watkins	14	True
8	Ilkay Gündogan	13	True
9	Alexandre Lacazette	13	True
10	Callum Wilson	12	True
11	Kelechi Iheanacho	12	True
12	Danny Ings	12	True
13	Chris Wood	12	True

Gambar 33. Pencilan menurut kriteria ketiga 3-sigma rule pada data jumlah gol di dataset EPL Goal Scorer

Deteksi pencilan menggunakan *Tukey's fences* memanfaatkan besaran rentang antarkuartil (*interquartile range*, IQR). Jika Q_1 menyatakan nilai kuartil pertama dari kumpulan data, lalu Q_3 merupakan nilai kuartil ketiganya. Maka, $IQR = Q_3 - Q_1$. Nilai IQR ini dapat dipakai untuk menentukan apakah suatu butir data merupakan pencilan dengan memanfaatkan kriteria sebagai berikut:

- x_i adalah pencilan jika $x_i < Q_1 - 1.5(IQR)$ atau $x_i > Q_3 + 1.5(IQR)$;
- x_i adalah pencilan ekstrem jika $x_i < Q_1 - 3(IQR)$ atau $x_i > Q_3 + 3(IQR)$.

Kriteria di atas dapat diimplementasikan dengan skrip sederhana berikut yang diterapkan pada kolom-kolom numerik dari *dataset* EPL Goal Scorer. Keluaran skrip dapat dilihat pada Gambar 34.

	player_name	goals	is_outlier	is_extreme_outlier					
0	Harry Kane	23	True	True	23	Edinson Cavani	10	True	True
1	Mohamed Salah	22	True	True	24	Anwar El Ghazi	10	True	True
2	Bruno Fernandes	18	True	True	25	Tomas Soucek	10	True	True
3	Son Heung-Min	17	True	True	26	Roberto Firmino	9	True	True
4	Patrick Bamford	17	True	True	27	Jesse Lingard	9	True	True
5	Dominic Calvert-Lewin	16	True	True	28	Riyad Mahrez	9	True	True
6	Jamie Vardy	15	True	True	29	Harvey Barnes	9	True	True
7	Ollie Watkins	14	True	True	30	Diogo Jota	9	True	True
8	Ilkay Gündogan	13	True	True	31	Che Adams	9	True	True
9	Alexandre Lacazette	13	True	True	32	James Ward-Prowse	8	True	False
10	Callum Wilson	12	True	True	33	Jarrod Bowen	8	True	False
11	Kelechi Iheanacho	12	True	True	34	Neal Maupay	8	True	False
12	Danny Ings	12	True	True	35	Gabriel Jesus	8	True	False
13	Chris Wood	12	True	True	36	Nicolas Pepe	8	True	False
14	Wilfried Zaha	11	True	True	37	Phil Foden	8	True	False
15	Marcus Rashford	11	True	True	38	Joe Willock	8	True	False
16	Sadio Mané	11	True	True	39	James Maddison	8	True	False
17	Gareth Bale	11	True	True	40	Stuart Dallas	8	True	False
18	Matheus Pereira	11	True	True	41	Jack Harrison	8	True	False
19	Pierre-Emerick Aubameyang	10	True	True	42	Bertrand Traoré	7	True	False
20	Michail Antonio	10	True	True	43	Jorginho	7	True	False
21	Christian Benteke	10	True	True	44	Rodrigo	7	True	False
22	Raheem Sterling	10	True	True	45	Richarlison	7	True	False
46	Ferrán Torres	7	True	False					
47	Mason Greenwood	7	True	False					
48	David McGoldrick	7	True	False					
49	Timo Werner	6	True	False					
50	Danny Welbeck	6	True	False					
51	Jack Grealish	6	True	False					
52	Tammy Abraham	6	True	False					
53	Gylfi Sigurdsson	6	True	False					
54	James Rodríguez	6	True	False					
55	Youri Tielemans	6	True	False					
56	Mason Mount	6	True	False					
57	Raphinha	6	True	False					

Gambar 34. Keluaran skrip pencari pencilan menurut kriteria Tukey.

7.4.4 Tabel Frekuensi

Bagian dari analisis data, khususnya untuk kolom-kolom yang bertipe nominal atau adalah menampilkan tabel frekuensinya. Dari Gambar 26 dan Gambar 27, diketahui bahwa terdapat 522 nilai unik untuk kolom `player_name` pada keseluruhan 522 baris data. Namun, dari kolom `team_title` dapat diketahui bahwa hanya ada 28 tim dalam tabel yang ternyata juga tidak cocok dengan keadaan sebenarnya, yakni ada 20 tim di English Premier League. Oleh sebab itu, data frekuensi mungkin dapat memberikan informasi lebih jauh dalam konteks ini.

Untuk mendapatkan data frekuensi, DataFrame memiliki fungsi `value_counts()`. Nilai tertinggi `value_counts()` menunjukkan modus dari kolom yang bersangkutan. Berikut skripnya untuk diterapkan pada kolom `team_title` dengan hasil pada Gambar

```
df_noid['team_title'].value_counts()
```

```

West Bromwich Albion      28
Everton                   28
Manchester United         27
Fulham                    27
Southampton               27
Sheffield United          27
Liverpool                 27
Wolverhampton Wanderers  27
Leicester                 27
Arsenal                   26
Brighton                  26
Newcastle United          26
Burnley                   25
Chelsea                   25
Tottenham                 24
Crystal Palace            24
Manchester City            24
Leeds                     23
Aston Villa               23
West Ham                  23
Everton,Southampton       1
Arsenal,West Bromwich Albion 1
Aston Villa,Chelsea        1
Chelsea,Fulham              1
West Bromwich Albion,West Ham 1
Arsenal,Newcastle United    1
Liverpool,Southampton      1
Arsenal,Brighton           1
Name: team_title, dtype: int64

```

Gambar 35. Keluaran fungsi `value_counts()` yang diterapkan pada kolom `team_title`.

Dari Gambar 35, terlihat bahwa ternyata West Bromwich Albion dan Everton memiliki 28 pemain yang tercatat di dalam data, sementara Leeds, Aston Villa, dan West Ham memiliki 23 pemain. Lalu, ada 8 nama tim yang tidak lazim, yakni terdiri dari komposisi dua nama tim. Penjelasan atas hal ini tidak dapat diperoleh dari datanya saja, namun dengan memanfaatkan pengetahuan terkait English Premier League, yakni bahwa dalam satu musim liga, seorang pemain bisa saja tergabung pada lebih dari satu tim. Hal ini terjadi ketika ada transfer atau peminjaman pemain di tengah musim antara kedua tim. Pada pemrosesan data selanjutnya (misalnya pada tahapan Data Preparation), data

pemain yang tergabung dalam lebih dari satu tim semusim dapat diperlakukan secara tersendiri.

7.4.5 Pengelompokan Data Berdasarkan Kolom

Analisis data juga dapat dilakukan dengan mengelompokkan data berdasarkan kolom-kolom tertentu. Misalnya, yang ingin diketahui bukan rerata jumlah gol yang dicetak setiap pemain secara keseluruhan, tetapi rerata jumlah gol yang dicetak per pemain untuk setiap tim. Untuk merealisasikannya, fungsi `DataFrame.groupby()` dapat dipergunakan. Keluaran fungsi tersebut sebenarnya adalah sebuah objek `DataFrameGroupBy` yang menyerupai `DataFrame` aslinya, namun sudah terkelompokkan sesuai parameter yang diberikan ke fungsi `groupby()`. Lalu, fungsi-fungsi statistika seperti rerata, simpangan baku, dll. dapat diterapkan pada objek `DataFrameGroupBy` seperti halnya pada `DataFrame` biasa, hanya saja agregasi diterapkan di masing-masing grup. Contohnya, jika fungsi rerata yang dipakai, maka rerata tersebut diterapkan untuk masing-masing grup.

Skrip berikut menghitung rerata gol yang dicetak per pemain untuk setiap tim, lalu mengurutkan hasilnya berdasarkan rerata gol tersebut. Hasil terlihat pada Gambar 36

```
df_noid.groupby('team_title')[['goals']]\n      .mean().sort_values(by='goals',ascending=False)
```


	goals
team_title	
Arsenal,Newcastle United	8.000000
Manchester City	3.208333
Aston Villa,Chelsea	3.000000
Liverpool,Southampton	3.000000
Everton,Southampton	3.000000
Tottenham	2.750000
Leeds	2.608696
Manchester United	2.518519
West Ham	2.478261
Leicester	2.370370
Liverpool	2.370370
Chelsea	2.240000
Aston Villa	2.130435
Arsenal	1.961538
Crystal Palace	1.625000
Everton	1.607143
Southampton	1.555556
Brighton	1.500000
Newcastle United	1.384615
Burnley	1.280000
Wolverhampton Wanderers	1.222222
West Bromwich Albion	1.178571
Chelsea,Fulham	1.000000
Fulham	0.925926
Sheffield United	0.666667
Arsenal,Brighton	0.000000
West Bromwich Albion,West Ham	0.000000
Arsenal,West Bromwich Albion	0.000000

Gambar 36. Rerata gol per pemain dari tim yang sama.

Data pada Gambar 36 menunjukkan bahwa rata-rata setiap pemain di tim Manchester City menyumbangkan lebih dari 3 gol bagi timnya, dan ini merupakan rerata kontribusi gol per pemain tertinggi di antara 20 tim yang tergabung dalam English Premier League musim tersebut. Di sisi lain, rerata kontribusi gol per pemain di tim Sheffield United kurang dari 1 gol, dan tim tersebut pada akhir musim terdegradasi ke divisi liga yang lebih rendah. Namun demikian, yang menempati baris paling atas pada Gambar 36 justru “tim” Arsenal+Newcastle United. Jika dicek silang dengan keluaran di Gambar 35, “tim” ini hanya terdiri dari 1 orang, yakni pemain Arsenal yang kemudian dipinjam oleh Newcastle United. Menariknya, pemain ini menyumbangkan 8 gol secara keseluruhan.

7.4.6 Analisis Korelasi

Analisis korelasi dilakukan pada data untuk mengetahui bagaimana hubungan dependensi antar dua buah kolom numerik pada data. Walaupun ada banyak besaran yang diusulkan untuk mengukur korelasi ini, satu yang paling sering dipakai (dan juga diimplementasikan sebagai fungsi di Pandas) adalah koefisien korelasi Pearson (Pearson's *correlation coefficient*) atau disingkat PCC.²¹ Nilai PCC berada dalam rentang -1 hingga 1, dan menggambarkan apakah satu variabel/kolom bergantung secara linier dengan variabel/kolom lainnya.

Andaikan sampel N buah data diberikan dengan dua buah atribut X dan Y , yakni x_i adalah nilai kolom X untuk data i , dan y_i adalah nilai kolom Y untuk data i . Kemudian, μ_x dan μ_y masing-masing adalah rerata nilai untuk kolom X dan Y , serta s_x dan s_y masing-masing adalah simpangan baku sampel (dengan koreksi Bessel) untuk data pada kolom X dan Y . Maka, PCC antara X dan Y , dinotasikan dengan r_{xy} , didefinisikan dengan formula:

$$r_{xy} = \frac{1}{(N-1)s_x s_y} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$$

Pembagi $N - 1$ adalah sesuai koreksi Bessel yang dipakai pada simpangan baku sampel.

Nilai r_{xy} dapat dimaknai sebagai derajat keterhubungan linier antara X dan Y dan selalu berada dalam rentang antara -1 dan 1. Jika $r_{xy} = 0$, maka dapat disimpulkan bahwa sama sekali tidak ada hubungan linier antara kedua kolom. Jika $r_{xy} = 1$, maka X dan Y berkorelasi linier positif secara sempurna, sedangkan jika $r_{xy} = -1$, maka X dan Y berkorelasi linier negatif secara sempurna. Secara umum, jika nilai r_{xy} positif, maka nilai kolom X dan Y meningkat secara bersama atau menurun secara bersama. Semakin kuat kecenderungan meningkat/menurun bersamanya, semakin mendekati 1 nilai r_{xy} -nya. Jika r_{xy} negatif, maka arah perubahan X dan Y berlawanan, yakni X meningkat ketika Y menurun, dan X menurun ketika Y meningkat.

Secara umum, ketika X dan Y independen, maka $r_{xy} = 0$, yakni tidak saling berkorelasi. Namun, PCC hanya sensitif terhadap dependensi linier antara dua variabel. Jadi, ketika $r_{xy} = 0$, maka belum tentu X dan Y independen: yang diketahui hanyalah bahwa tidak ada korelasi linier antara keduanya, dan bukan independensi antara keduanya.

²¹ https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

Di Pandas DataFrame, nilai PCC dapat diperoleh secara langsung antara setiap pasangan kolom numerik pada data, yakni dengan skrip berikut:

```
df_noid.corr()
```

Hasilnya ada pada Gambar 37. Hasil perintah tersebut adalah sebuah matriks simetrik dengan jumlah baris dan kolom yang sama dengan banyaknya kolom numerik pada DataFrame aslinya.

	games	time	goals	xG	assists	xA	shots	key_passes	yellow_cards	red_cards	npg	npG	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	0.618754
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	0.726598
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	0.290990
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	0.290990
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	0.473254
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	0.547983
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	0.448197
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	0.618754
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	0.562467
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	0.167660
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	0.284135
npG	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	0.273090
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	0.802073
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	1.000000

Gambar 37. Nilai koefisien korelasi Pearson yang dikeluarkan oleh fungsi corr()

Beberapa hal dapat dilihat pada hasil tersebut. Misalnya, terlihat bahwa terdapat korelasi positif yang sangat kuat, yakni 0.945 antara kolom **games** dan **time** yang memang wajar karena **time** menunjukkan total menit dalam musim seorang pemain bermain di dalam suatu pertandingan. Jika setiap pemain terlibat di suatu pertandingan secara penuh (tidak ada pergantian pemain, atau kejadian berhenti karena cedera, atau diberikan kartu merah oleh wasit), maka PCC antara kedua kolom tersebut bisa jadi akan bernilai 1.

Korelasi positif yang kuat juga terlihat antara jumlah gol dengan jumlah tembakan ke gawang (**shots**), serta antara jumlah *assist* dengan jumlah umpan krusial (*key passes*). Sementara itu, tidak ada pasangan kolom yang berkorelasi negatif kuat. Yang ada adalah kolom-kolom yang tidak berkorelasi linier, misalnya antara jumlah gol dengan jumlah kartu kuning atau kartu merah. Lalu jumlah kartu kuning dengan jumlah kartu merah sendiri juga tidak memiliki korelasi yang kuat. Hal ini menunjukkan bahwa ketika seorang pemain dijatuhi hukuman kartu kuning oleh wasit, tidak secara signifikan

membuat pemain tersebut menjadi jauh lebih mungkin untuk dijatuhi hukuman kartu merah.

Tugas Dan Proyek Pelatihan (Tugas Harian)

Tugas 1

Anda diminta untuk melakukan pengambilan data pada kaggle dengan cara menggunakan tahapan-tahapan yang sudah ada di slide. Anda hanya perlu melakukan download 2 dataset bebas menggunakan tahapan-tahapan tersebut. Screenshot terminal (termasuk kaggle.json) dan list isi folder data dimasukkan ke dalam dokumen yang dikumpulkan.

<https://www.kaggle.com/docs/api>

Filename pengumpulan : tugas-6-1-<id-peserta>.docx

Pastikan filename yang dikumpulkan sesuai dengan format untuk mempermudah admin mengelola jawaban anda.

Tugas 2

<https://github.com/hackathonBI/CCS>

Buatlah file ipynb untuk dapat melakukan data understanding berikut ini :

1. Buatlah table list Top 5 Customers yang memiliki nilai transaksi paling banyak menggunakan dataframe pada python !
2. Buatlah list Top 5 Gas stations yang memiliki nilai transaksi paling banyak menggunakan dataframe pada python!
3. Buatlah list Top 5 Jenis produk yang memiliki nilai transaksi paling banyak menggunakan dataframe pada python!
4. Buatlah deskripsi statistik untuk masing masing hari (23,24,25,26) pada dataset transaction_1k!
5. Carilah waktu terbaik (hari dan jam) dimana paling banyak user gas station melakukan transaksi !
6. Dari segi bisnis understanding, tujuan utama dari analisis data customer ini adalah?- Masukkan jawaban anda pada text di ipynb

Filename pengumpulan : tugas-6-2-<id-peserta>.ipynb

Pastikan filename yang dikumpulkan sesuai dengan format untuk mempermudah admin mengelola jawaban anda.

Link Referensi Modul Enam

Link Pertanyaan Modul Enam

Bahan Tayang
Power Point

Link room Pelatihan dan Jadwal live sesi bersama instruktur
Zoom

Penilaian
Komposisi penilaian Tugas Data Understanding 1: Nilai 100

Target Penyelesaian Modul Enam
1 hari / sampai 6 JP



KOMINFO

Badan Penelitian dan Pengembangan SDM
Kementerian Komunikasi dan Informatika