**Apa itu telaah data?**

Telaah data merujuk pada proses pengumpulan, pembersihan, eksplorasi, dan analisis data untuk mendapatkan pemahaman yang lebih yang terkandung dalam data tersebut. Tujuan dari telaah data adalah mendukung pengambilan keputusan berdasarkan bukti yang ditemukan dalam data.

**Mengapa perlu telaah data?**

1. Data dari masing-masing sumber belum tentu dapat langsung dipakai karena:

* maksud dan tujuan data berbeda-beda
* keadaan asal terpisah-pisah atau justru terintegrasi secara ketat.
* tingkat kekayaan (richness) berbeda-beda
* tingkat keandalan (reliability) berbeda-beda

1. Data understanding memberikan gambaran awal tentang:

* kekuatan data
* kekurangan dan batasan penggunaan data
* tingkat kesesuaian data dengan masalah bisnis yang akan dipecahkan
* ketersediaan data (terbuka/tertutup, biaya akses, dsb.)

**Bentuk Data**

Data dapat memiliki berbagai bentuk, diantaranya:

* Spreadsheet(excel, csv, dll)
* Database(SQL, Oracle, dll)
* Text file(txt, doc, pdf, dll)
* Multimedia documents(audio, video, gambar, dll)

**Sumber Data**

1. Sumber Internal

* Data yang dihasilkan oleh perusahaan sendiri
* Data yang dihasilkan oleh perusahaan lain yang terkait dengan perusahaan

1. Sumber Eksternal

* Repositori data publik
* Halaman web publik

**Daftar Sumber Data Daring**

* [Portal Satu Data Indonesia](https://data.go.id/)
* [Portal Data Jakarta](https://data.jakarta.go.id/)
* [Portal Data Bandung](http://data.bandung.go.id/)
* [Badan Pusat Statistik](https://www.bps.go.id/)
* [Badan Informasi Geospasial](https://tanahair.indonesia.go.id/)
* [UCI Machine Learning repository](https://archive.ics.uci.edu/ml/index.php)
* [Kaggle](https://www.kaggle.com/datasets)
* [World Bank Open Data](https://data.worldbank.org/)

**Tipe Data**

**Bedasarkan sifat**

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

**Bedasarkan waktu**

* **Data Cross Section**, adalah data yang menunjukkan titik waktu tertentu. Contohnya laporan keuangan per 31 Desember 2020, data pelanggan PT.Data Indah bulan mei 2004, dan lain sebagainya.
* **Data Time Series / Berkala**, adalah data yang datanya menggambarkan sesuatu dari waktu ke waktu atau periode secara historis. Contoh data time series adalah data perkembangan nilai tukar dollar amerika terhadap rupiah tahun 2016 - 2020

**Pengambilan Data**

* Pengambilan data secara manual.
* Pengambilan data melalui API
  + Contoh melalui API Kaggle (pip install kaggle)(kaggle datasets download -d mauryansshivam/paytm-revenue-users-transactions)
  + Contoh melalui API Portal Data Bandung (http://data.bandung.go.id/index.php/portal/api/1db4a0cc-dc0f-4362-91f1-d0ad1b4bce98)
* Pengambilan data melalui akses langsung ke basis data relasional yang ada.
* Pengambilan data melalui web scraping.
  + Contoh pengambilan data melalui web scraping kometar video youtube

**Menampilkan Data**

Method head() dan tail() pada DataFrame membantu kita menampilkan 5 beberapa baris pertama/terakhir dari data yang kita muat. Dengan memberikan argumen pada method tersebut kita juga bisa mengatur jumlah data yang ditampilkan

# 5 baris pertama

df.head()

# 5 baris terakhir

df.tail()

# 10 baris pertama

df.head(10)

**Melihat Tipe Data**

Method dtypes pada DataFrame membantu kita menampilkan tipe data dari setiap kolom pada data yang kita muat. Jika terdapat tipe data yang tidak tepat maka sebaiknya dilakukan pengecekan nialai pada kolom tersebut. Jika memang sudah benar, maka kita bisa mengubah tipe data tersebut menjadi tipe data yang tepat.

# Melihat tipe data

df.dtypes

# Mengubah tipe data menjadi int

df['nama\_kolom'] = df['nama\_kolom'].astype('int')

**Deskripsi statistik data**

Method describe() pada DataFrame membantu kita menampilkan deskripsi statistik dari data yang kita muat. Deskripsi statistik yang ditampilkan adalah deskripsi statistik untuk kolom-kolom dengan tipe data numerik. Jika terdapat kolom dengan tipe data selain numerik, maka deskripsi statistik tersebut tidak akan ditampilkan. Untuk tetap menampilkan deskripsi statistik dari kolom non-numerik, kita bisa menambahkan argumen include='all' pada method describe().

# Deskripsi statistik data

df.describe()

# Deskripsi statistik data termasuk kolom non-numerik

df.describe(include='all')

**Fungsi statistik dalam Pandas**

Pandas menyediakan fungsi statistik untuk menghitung nilai rata-rata, median, standar deviasi, dan variansi. Fungsi-fungsi tersebut adalah:

| **Fungsi** | **Keterangan** |
| --- | --- |
| count | Jumlah observasi non-NULL |
| sum | Jumlah |
| mean | Rata-rata |
| mad | Deviasi absolut rata-rata |
| median | Nilai tengah |
| min | Nilai minimum |
| max | Nilai maksimum |
| mode | Modus (nilai yang paling sering muncul) |
| abs | Nilai absolut (nilai mutlak) |
| prod | Hasil kali dari nilai-nilai |
| quantile | Kuantil sampel (nilai pada persentil), kuartil pertama = quantile(0.25) |
| std | Standar deviasi |
| var | Varians |
| sem | Standar error mean (standar error dari rata-rata) |
| skew | Skewness (kecondongan distribusi data) |
| kurt | Kurtosis (tingkat “tumpul” atau “tajam” distribusi data) |
| cumsum | Akumulasi jumlah |
| cumprod | Akumulasi hasil kali |
| cummax | Nilai maksimum akumulasi |
| cummin | Nilai minimum akumulasi |

**3.8.6 Mentukan Outlier**

Outlier adalah data yang memiliki nilai ekstrim dibandingkan dengan data lainnya. Outlier dapat mempengaruhi hasil analisis data sehingga perlu ditangani dengan baik. Untuk mengetahui apakah suatu data merupakan outlier atau bukan, kita bisa menggunakan beberapa metode, diantaranya:

* Tukey Fences (IQR)
  + **Plus**: IQR adalah metode yang tahan terhadap nilai ekstrim dan lebih baik digunakan pada data dengan distribusi yang condong (skewed) karena mengandalkan kuartil (quartiles) yang tidak terpengaruh oleh ekstrem nilai.
  + **Minus**: IQR tidak dapat digunakan pada data dengan distribusi normal karena mengandalkan kuartil (quartiles).
* Z-Score
  + **Plus**: Z-score dapat digunakan pada data dengan distribusi normal karena mengandalkan mean dan standard deviasi.
  + **Minus**: Z-score tidak tahan terhadap nilai ekstrim.
* Metode Hampiran Berbasis Probabilitas (Contoh: grubb’s test)
  + **Plus**: Metode hampiran berbasis probabilitas seperti Grubbs’ Test dapat memberikan kepercayaan statistik dalam menentukan apakah sebuah data benar-benar outlier atau hanya perbedaan alami dalam distribusi data.
  + **Minus**: Memerlukan asumsi bahwa data harus berdistribusi normal. Selain itu, metode ini hanya cocok untuk mengidentifikasi satu outlier dalam satu arah (outlier yang memiliki nilai ekstrim di atas atau di bawah rata-rata).

**3.8.7 Nilai Unik**

Method unique() pada Pandas digunakan untuk mengetahui nilai unik dari suatu kolom. Method ini mengembalikan nilai unik dari suatu kolom dalam bentuk array. Method value\_counts() pada Pandas digunakan untuk menghitung berapa kali suatu nilai muncul dalam suatu kolom. Method ini mengembalikan Series yang berisi frekuensi setiap nilai yang muncul dalam suatu kolom.

# Penggunaan unique

df["nama\_kolom"].unique()

# Penggunaan value\_counts

df["nama\_kolom"].value\_counts()

**Analisis dengan groupby**

Method groupby() pada Pandas digunakan untuk melakukan grouping berdasarkan kolom tertentu. Method ini mengembalikan objek DataFrameGroupBy.

# Penggunaan groupby dengan fungsi mean

df.groupby("kolom1")["kolom2"].mean()

# menghitung statistik deskriptif dari setiap kelompok data

df.groupby("kolom1")["kolom2"].describe()

# agregasi kustom dengan fungsi agg

df.groupby("kolom1")["kolom2"].agg([sum, min, max])

#Menggunakan operasi agregasi kustom (menghitung rasio nilai maksimum dan minimum dalam setiap kelompok

df.groupby("kolom1")["kolom2"].agg(lambda x: x.max() / x.min())

**Korelasi**

Korelasi adalah salah satu metode statistik yang digunakan untuk mengetahui seberapa besar hubungan antara satu variabel dengan variabel lainnya. Korelasi memiliki nilai berkisar antara -1 hingga 1. Nilai 1 menunjukkan hubungan yang sempurna, nilai -1 menunjukkan hubungan yang sempurna pula dengan arah yang berlawanan, dan nilai 0 menunjukkan tidak ada hubungan antara kedua variabel tersebut. Kita dapat menggunakan methon corr() pada Pandas untuk menghitung korelasi dari setiap pasang kolom pada suatu DataFrame.

# Menghitung korelasi

df.corr()

# Terdapat 3 pilihan metode perhitungan korelasi, yaitu:

# pearson (default), kendall, dan spearman

df.corr(method="kendall")