# Exploratory Data Analysis (EDA)

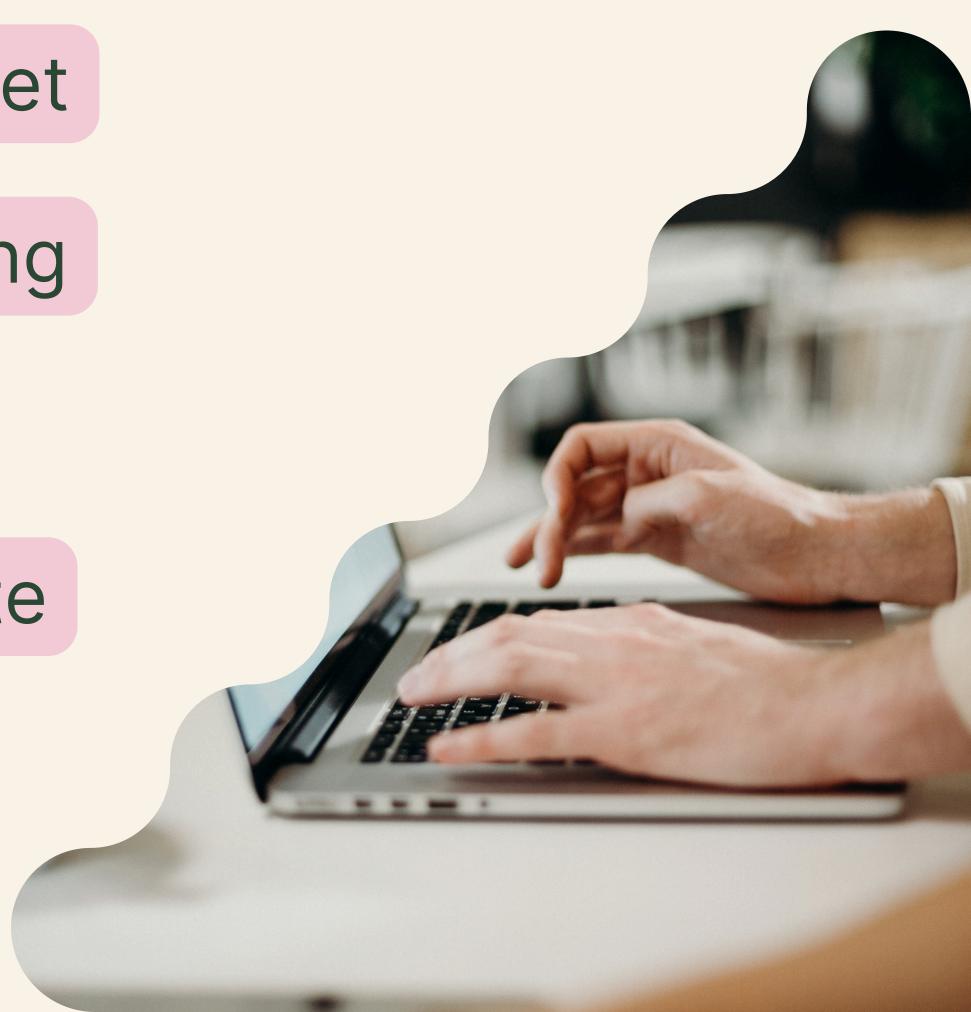
Widia Mulya Hartanti

Portofolio

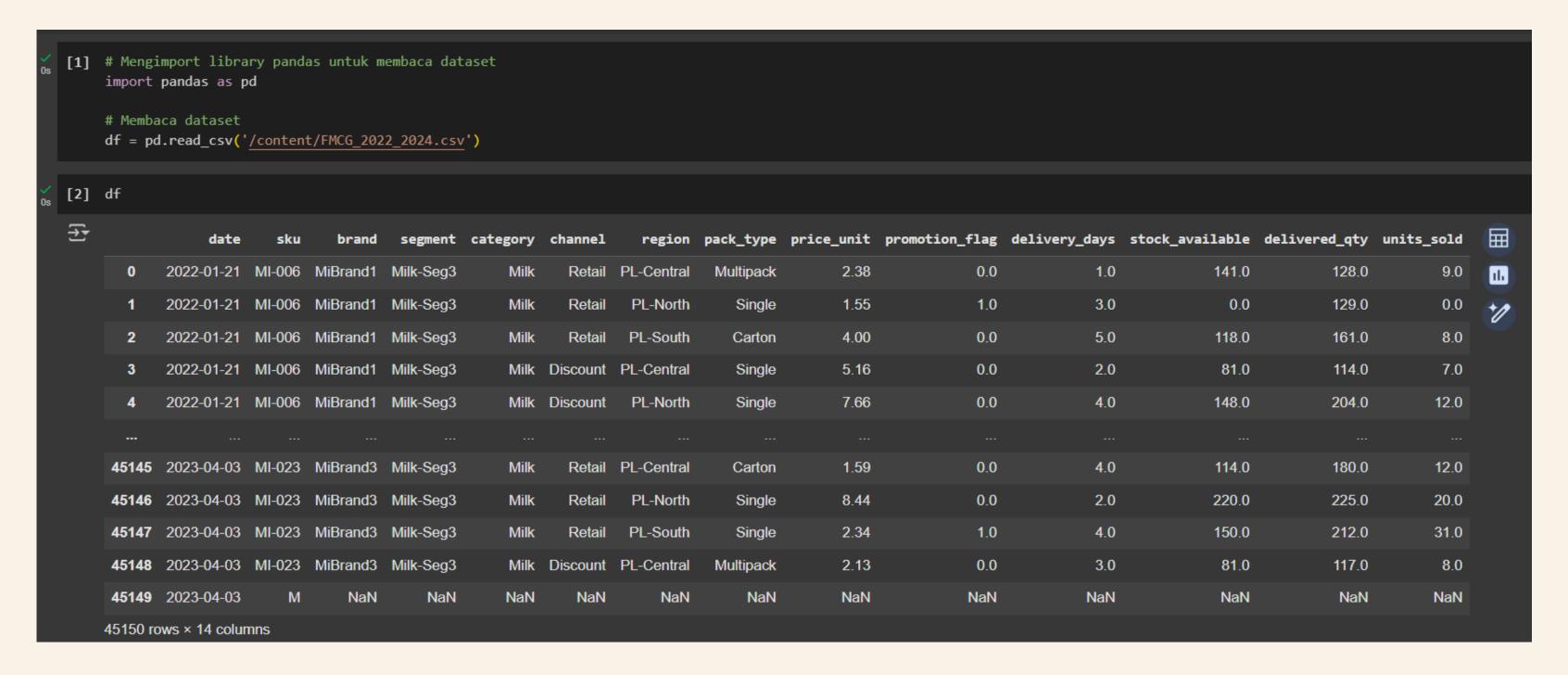
01 - Memasukkan dataset

02 - Mengecek data yang hilang (missing value)

03 - Mengecek dan mengatasi data duplicate



#### 01 - Memasukkan dataset

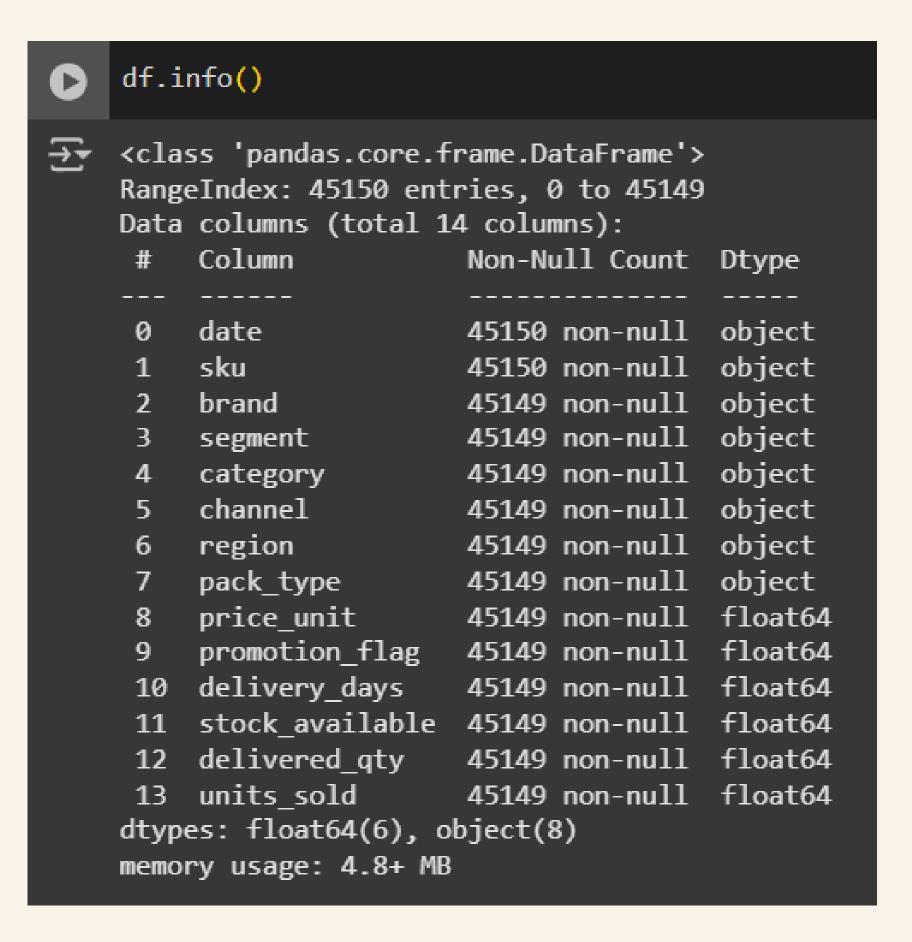


Sumber: https://www.kaggle.com/datasets/beatafaron/fmcg-daily-sales-data-to-2022-2024/data

#### Ringkasan data

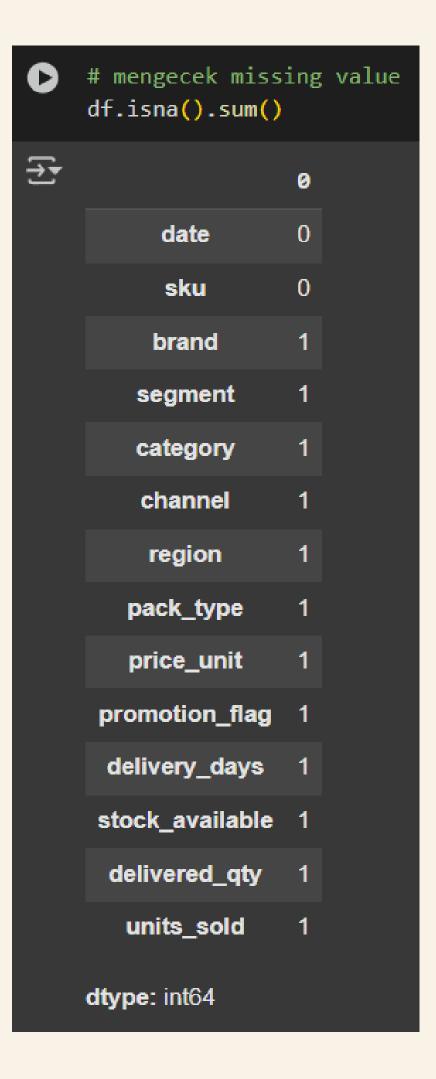
Menggunakan syntax info() untuk memahami struktur dan konten DataFrame dalam membantu eksplorasi dan persiapan data.

- DataFrame terdiri dari 45150 entri dengan indeks dari 0 sampai 45149.
- 8 kolom data bertipe object: date, sku, brand, segment, category, channel, region, dan pack\_type.
- 6 kolom data bertipe float64: price\_unit, promotion\_flag, delivery\_days, stock\_available, delivered\_qty, dan units\_sold.



#### 02 - Mengecek missing value

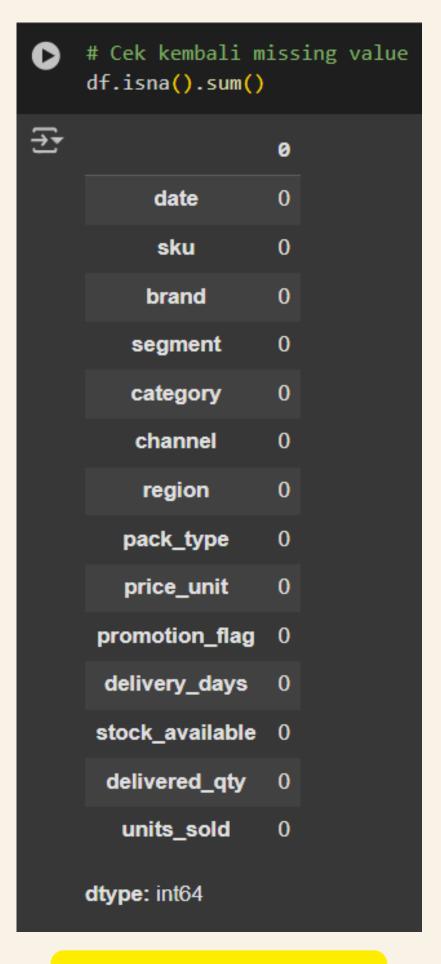
Berdasarkan informasi tersebut, didapatkan beberapa data yang hilang pada kolom brand, segment, category, channel, region, pack\_type, price\_unit, promotion\_flag, delivery\_days, stock\_available, delivered\_qty, dan units\_sold



### Mengatasi missing value

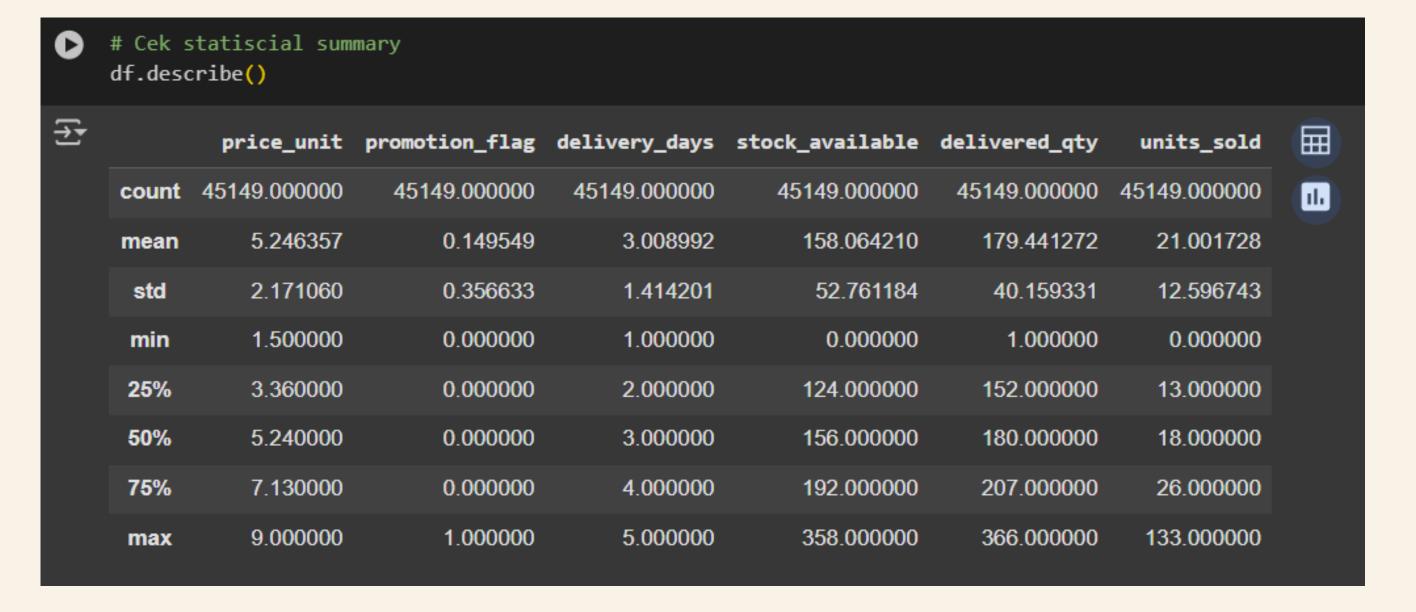
```
# Mengatasi missing value
for column in df.columns:
    if df[column].dtype == 'object':
        # Jika kolom bertipe object, isi dengan mode
        df[column].fillna(df[column].mode()[0], inplace=True)
    else:
        # Jika kolom bertipe numerik, isi dengan mean
        df[column].fillna(df[column].mean(), inplace=True)
```

Untuk mengatasi missing value di mana data bertipe object akan diisi dengan modus/nilai yang sering muncul. Jika data bertipe numerik akan diisi dengan mean/nilai rata-rata.



Hasil penyelesaikan missing value

Observasi data



Output syntax df.describe() menunjukkan bahwa data di atas memiliki nilai minimum dan maksimum untuk semua kolom yang masuk akal. Kolom delivery\_days memiliki distribusi normal sementara kolom price\_unit, stock\_available, units\_sold, dan delivered\_qty memiliki skewness yang jelas yaitu right-skewed. Kolom promotion\_flag adalah kolom boolean/biner karena nilainya 0 atau 1, sehingga tidak perlu menyimpulkan kesimetrisannya, hanya perlu memeriksa balance level.

## 03 - Mengecek data duplicate

```
# Mengecek apakah ada duplicate di seluruh kolom
check_duplicate = df.duplicated().sum()

print(f"Jumlah data yang duplikat = {check_duplicate}")

Jumlah data yang duplikat = 0
```

Berdasarkan gambar tersebut diketahui bahwa tidak ada data duplikat

## Mengatasi data duplicate

```
# Handling duplicate
df = df.drop_duplicates()

[] # Mengecek duplicate setelah di-handle
handle_duplicate = df.duplicated().sum()

print(f"Jumlah data yang duplikat = {handle_duplicate}")
```

Data duplicate dapat diatasi dengan syntax tersebut, di mana keseluruhan kolom data akan dicek lalu menghapus data yang duplikat.

## Thanks

Widia Mulya Hartanti