Exploratory Data Analysis (EDA)

Data Science
AMIKOM COMPUTER CLUB 2023/2024

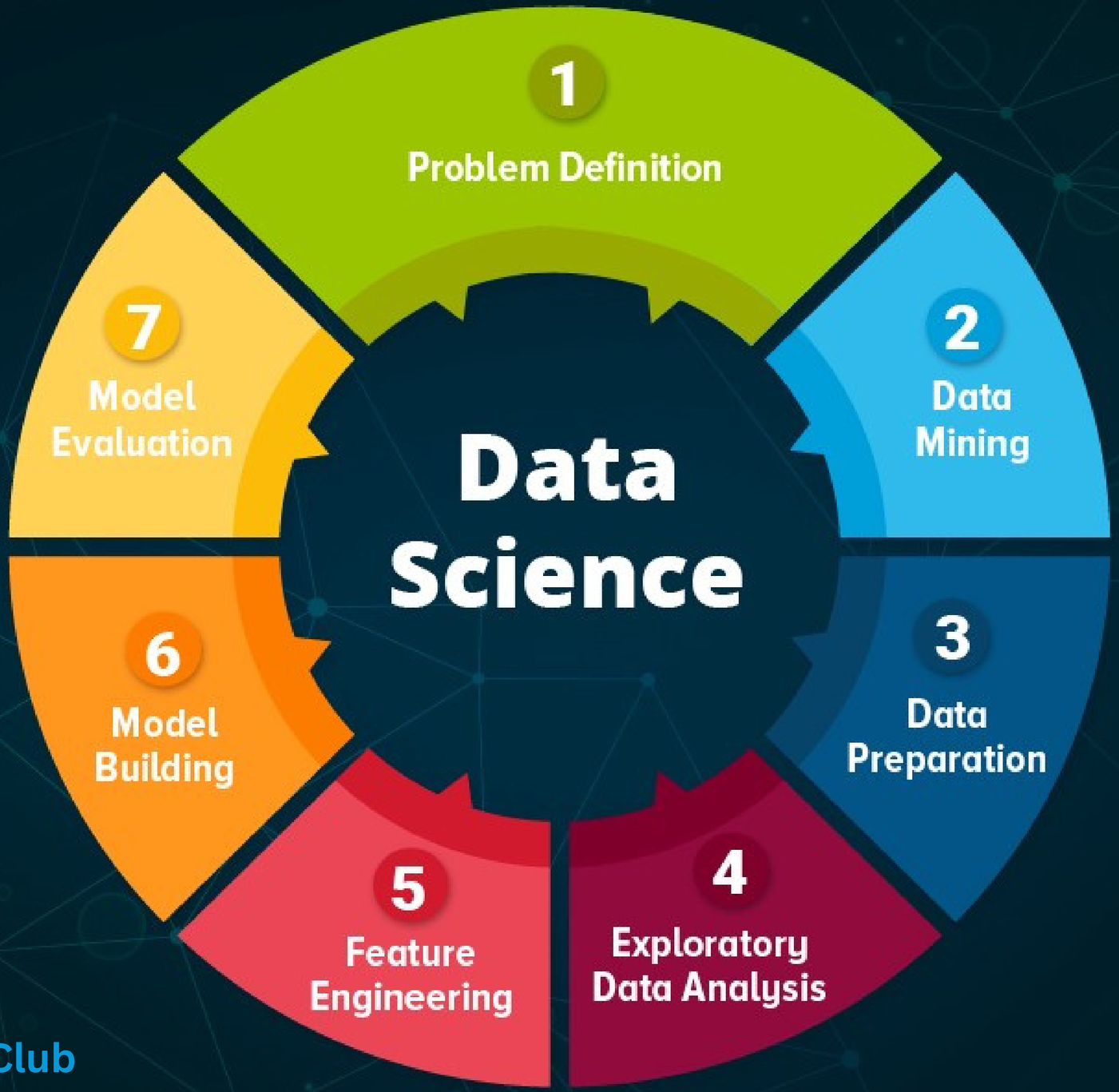
× Python

Apa itu EDA?

Exploratory Data Analysis (EDA) adalah bagian proses data science. EDA menjadi tahap awal yang penting. Tujuannya adalah untuk **memahami karakteristik utama dari kumpulan data.** Berikut beberapa hal yang dilakukan EDA :

1. **Menyimpulkan karekteristik data**
2. **Membuat visualisasi data**

EDA dilakukan sebelum masuk ke tahap pemodelan atau pengujuan hipotesis.



Amikom Computer Club

Perlakuan EDA

Dalam prakteknya , pemahaman konteks data perlu diperhatikan, karena akan menjawab masalah masalah besar.

→ Pada umumnya EDA dilakukan dengan beberapa cara,

- **Univariat Analysis**- analisis deskriptif dengan satu variabel.
- **Bivariat Analysis**- analisis relasi dengan dua variabel yang biasanya dengan target variabel.
- **Multivariat Analysis** - analisis yang menggunakan lebih dari atau sama dengan tiga variabel..

Tahapan EDA

Berikut adalah tahapan pada EDA:

1. **Memahami Data**
- Memahami Konteks, Memeriksa Struktur Data, Menentukan Pernyataan Penelitian
2. **Statistika Deskriptif**
- Menganalisis statistika dasar, Menganalisis Distribusi Data, Menemukan Outliers
3. **Visualisasi Data**
- Membuat visualisasi data berdasarkan tujuan analisis
4. **Menemukan Pola dan Anomali**
- Mencari pola, Menemukan Anomali, Mempelajari Outlier
5. **Menarik Kesimpulan**
- Menarik Kesimpulan, Membuat hipotesis, Merancang langkah selanjutnya

Observasi

→ Observasi dalam EDA adalah langkah awal yang pentih dalam memahami karakteristik adat sebelum melakukan analisis lebih lanjut. Hal ini dilakukan dengan tujuan **mengenali jenis data, mengidentifikasi nilai-nilai ekstrim, dan memahami pola-pola umum** dalam dataset.

Melakukan **observasi** data dapat membantu kita untuk **memahami karakteristik data** secara lebih baik dan **memberikan informasi penting dalam pengambilan keputusan.**

Observasi

Beberapa hal yang perlu diperhatikan dalam melakukan observasi data meliputi :

→ **Jenis Data** : Melihat tipe data variabel, apakah variabel tersebut bersifat numerik atau kategorikal.

→ **Rentang Nilai Variabel** : Melihat rentang nilai dari setiap variabel numerik dalam dataset.

→ **Distribusi Data** : Melihat distribusi data dalam setiap variabel numerik, seperti apakah distribusi tersebut normal, skewed atau bimodal.

→ **Nilai-nilai Ekstrim** : Melihat nilai-nilai yang ekstrim, seperti outlier dan missing value dalam dataset.

Korelasi

Salah satu teknik yang dapat digunakan untuk menemukan pola dalam sebuah data adalah dengan menghitung **korelasi** dari variabel(feature).

→ **Korelasi** adalah teknik yang digunakan untuk mengukur hubungan antara dua variabel dalam sebuah dataset. Hal ini dapat membantu analis untuk memahami hubungan antara variabel-variabel tersebut dan memperoleh wawasan tentang bagaiman variabel-variabel tersebut mempengaruhi satu sama lain.

Di Python, kita apat menghitung korelasi antara dua variabel menggunakan library Numpy atau Pandas. **Korelasi** dapat dihitung menggunakan **koefisien korelasi Pearson, Spearman, atau Kendall**

Uji Peearson Product Moment

→ **Uji Person Product Moment** adalah satu dari beberapa jenis uji korelasi yang digunakan untuk mengetahui derajat keeratan hubungan 2 variabel yang berskala interval atau rasio, dimana dengan uji ini akan mengembalikan nilai koefisen korelasi yang nilainya berkisar antara **-1, 0, dan 1. Nilai -1** artinya terdapat **korelasi negatif yamg sempurna**, **0** artinya **tidak ada korelasi**, dan **nilai 1** berarti ada **korelasi positif yang sempurna**.

Interval Koefisien	Tingkat Hubungan
0,00 – 0,199	Sangat rendah
0,20 – 0,399	Rendah
0,40 – 0,599	Cukup
0,60 – 0,799	Kuat
0,80 – 1,000	Sangat kuat

Rumus Korelasi Product Moment

$$r_{xy} = \frac{n\sum x_i y_i - (\sum x_i)(\sum y_i)}{\sqrt{(n\sum x_i^2 - (\sum x_i)^2)(n\sum y_i^2 - (\sum y_i)^2)}}$$

Dimana:

r_{xy} = korelasi antara x dengan y

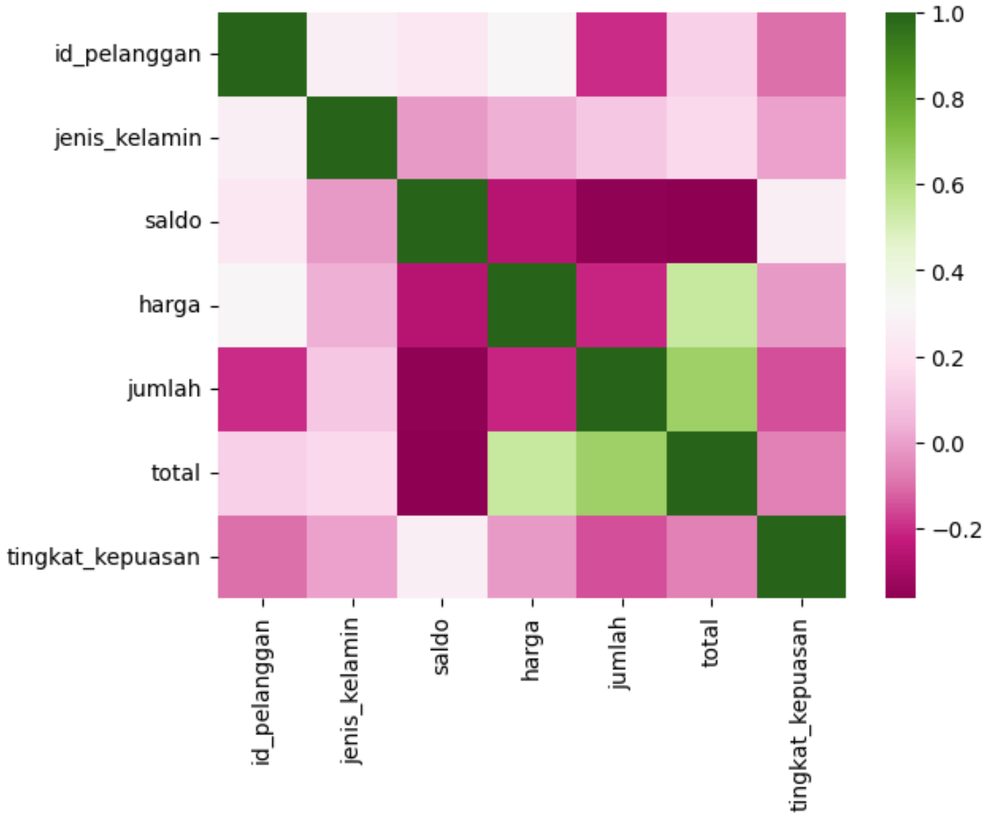
x_i = nilai x ke-i

y_i = nilai y ke-i

n = banyaknya nilai

(Sugiyono, 2011: 228)

Visualisasi Korelasi Heatmap

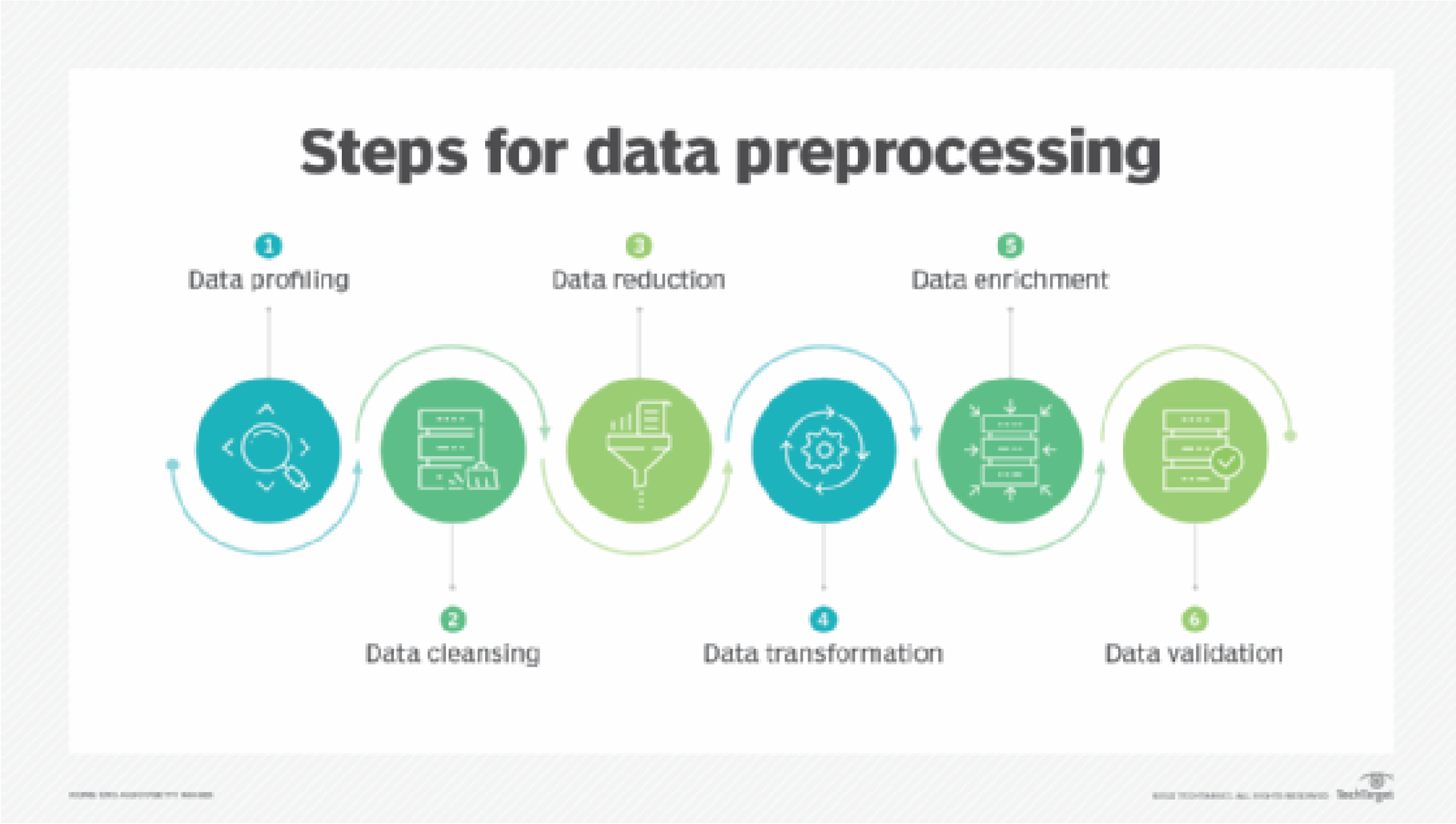


Preprocessing Data

→ **Prepocessing** data adalah tahap penting dalam **EDA (Exploratory Data Analysis).** Tujuannya adalah untuk mempersiapkan data mentah sebelum dianalisis dan membuatnya lebih siap untuk diinterpretasikan. Preprocessing data melibatkan beberapa teknik seperti **membersihkan data, menghilangkan noise, mengisi data yang hilang, dan mengubah format data.**

Teknik preprocessing data ini dapat membantu analis dalam mempersiapkan data sebelum dilakukan analisis secara lebih mendalam dan detail. Dengan menggunakan teknik preprocessing data yang tepat, data dapat dipersiapkan dengan lebih baik, sehingga hasil analisis dapat lebih akurat dan interpretasinya dapat lebih mudah dipahami.

Preprocessing Data



Removing Duplicates

1. **Discovering Duplicates**

→ Baris duplikat adalah baris yang telah didaftarkan lebih dari satu kali.

Untuk menemukan duplikat, kita bisa menggunakan **duplicated()** metode.

duplicated() mengembalikan nilai boolean untuk setiap baris

	Durasi	Tanggal	Pulsa	Maxpulse	Kalori
0	60	'2020/12/01'	110	130	409.1
1	60	'2020/12/02'	117	145	479.0
2	60	'2020/12/03'	103	135	340,0
3	45	'2020/12/04'	109	175	282,4
4	45	'2020/12/05'	117	148	406.0
5	60	'2020/12/06'	102	127	300,0
6	60	'2020/12/07'	110	136	374.0
7	450	'2020/12/08'	104	134	253,3
8	30	'2020/12/09'	109	133	195.1
9	60	'2020/12/10'	98	124	269.0
10	60	'2020/12/11'	103	147	329,3
11	60	'2020/12/12'	100	120	250,7
12	60	'2020/12/12'	100	120	250,7
13	60	'2020/12/13'	106	128	345.3
14	60	'2020/12/14'	104	132	379,3
15	60	'2020/12/15'	98	123	275.0
16	60	'2020/12/16'	98	120	215.2
17	60	'2020/12/17'	100	120	300,0
18	45	'2020/12/18'	90	112	Tidak
19	60	'2020/12/19'	103	123	323.0

Removing Duplicates

2. Removing Duplicates

→ Untuk menghapus duplikat, gunakan **drop_duplicateds()**.

```
df.drop_duplicates(inplace = True)
```

	Durasi	Tanggal	Pulsa	Maxpulse	Kalori
0	60	'2020/12/01'	110	130	409.1
1	60	'2020/12/02'	117	145	479.0
2	60	'2020/12/03'	103	135	340,0
3	45	'2020/12/04'	109	175	282,4
4	45	'2020/12/05'	117	148	406.0
5	60	'2020/12/06'	102	127	300,0
6	60	'2020/12/07'	110	136	374.0
7	450	'2020/12/08'	104	134	253,3
8	30	'2020/12/09'	109	133	195.1
9	60	'2020/12/10'	98	124	269.0
10	60	'2020/12/11'	103	147	329,3
11	60	'2020/12/12'	100	120	250,7
12	60	'2020/12/12'	100	120	250,7
13	60	'2020/12/13'	106	128	345.3
14	60	'2020/12/14'	104	132	379,3
15	60	'2020/12/15'	98	123	275.0
16	60	'2020/12/16'	98	120	215.2
17	60	'2020/12/17'	100	120	300,0
18	45	'2020/12/18'	90	112	Tidak
19	60	'2020/12/19'	103	123	323.0

Cleaning Empty Cell

- Empty Cell

→ Sel kosong berpotensi memberikan hasil yang salah saat kita menganalisis data.

- Remove Rows

→Salah satu cara mengatasi sel kosong adalah dengan menghilangkan baris yang berisi sel kosong.

Penting!

Secara dasar, **dropna()** metode ini mengembalikan DataFrame baru, dan tidak akan mengubah yang asli. Jika kita ingin mengubah DataFrame asli, gunakan inplace() = True argument :

```
import pandas as pd

df = pd.read_csv('data.csv')

new_df = df.dropna()

print(new_df.to_string())
```

Cleaning Empty Cell

Penting!
dropna(inplace = True) TIDAK akan mengembalikan DataFrame baru, tetapi akan menghapus semua baris yang berisi nilai NULL dari DataFrame asli.

```
import pandas as pd

df = pd.read_csv('data.csv')

df.dropna(inplace = True)

print(df.to_string())
```

Data Frame Isnull

→ Metode ini **isnull()** mengembalikan objek DataFrame yang semua nilainya diganti dengan nilai Boolean True untuk nilai NULL, dan sebaliknya False.

```
dataframe.isnull()
```

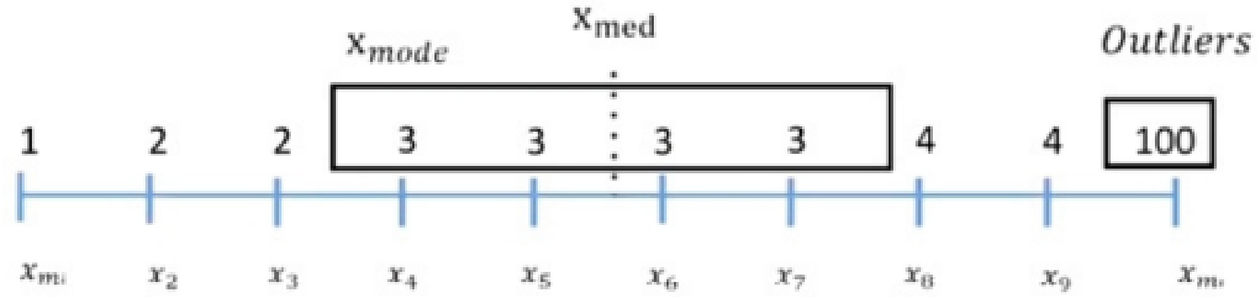
```
import pandas as pd

df = pd.read_csv('data.csv')

newdf = df.isnull()
```

Outliers

→ Outlier adalah nilai yang menyimpang dari nilai data data yang lain. Outliners dapat mengganggu nilai ukuran pemusatan dan penyebaran sehingga menjadi bias dan tidak representatif.

Pengaruh Outliers pada Data

→ Outlier dapat menyebabkan analisis yang tidak akurat atau tidak representatif karena mereka dapat mempengaruhi nilai rata-rata, standar deviasi, dan analisis lain yang didasarkan pada data keseluruhan.



$$\bar{x} = \frac{(1) + (2\,x2) + (3x3) + (4x3) + (100)}{10}$$
$$= 12.6$$

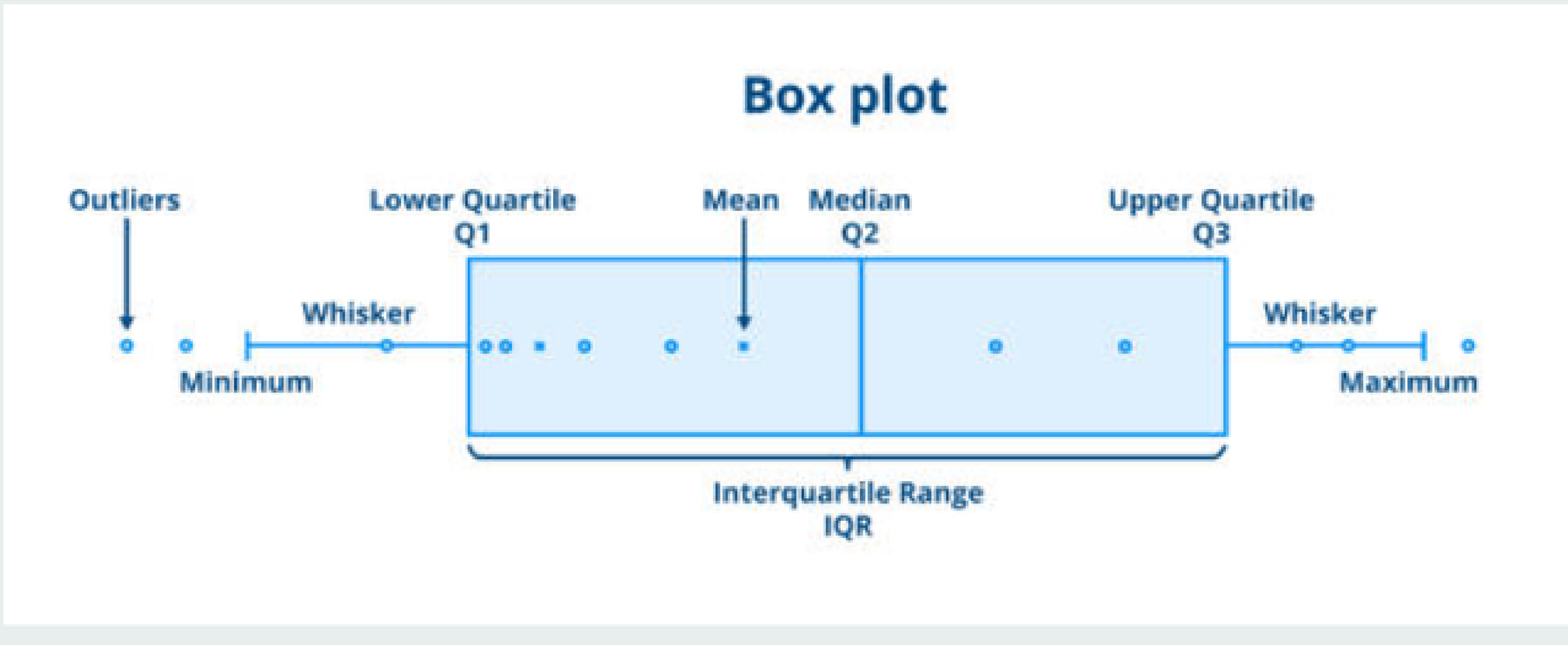
Cara Mendeteksi Outliners

1. **Mendapatkan nilai statistika lima serangkai**

- Minimum: Nilai terkecil dalam set data.
- Kuartil Pertama (Q1): Nilai yang membagi data menjadi dua bagian, di mana 25% data bernilai lebih kecil dari nilai Q1 dan 75% bernilai lebih besar dari nilai Q1.
- Median: Nilai tengah dalam set data. Setengah data bernilai lebih kecil dari median dan setengah data lagi bernilai lebih besar dari median.
- Kuartil Ketiga (Q3): Nilai yang membagi data menjadi dua bagian, di mana 75% data bernilai lebih kecil dari nilai Q3 dan 25% bernilai lebih besar dari nilai Q3.
- Maksimum: Nilai terbesar dalam set data. Kelima nilai statistik tersebut dapat digunakan untuk membuat boxplot dan melihat outlier.

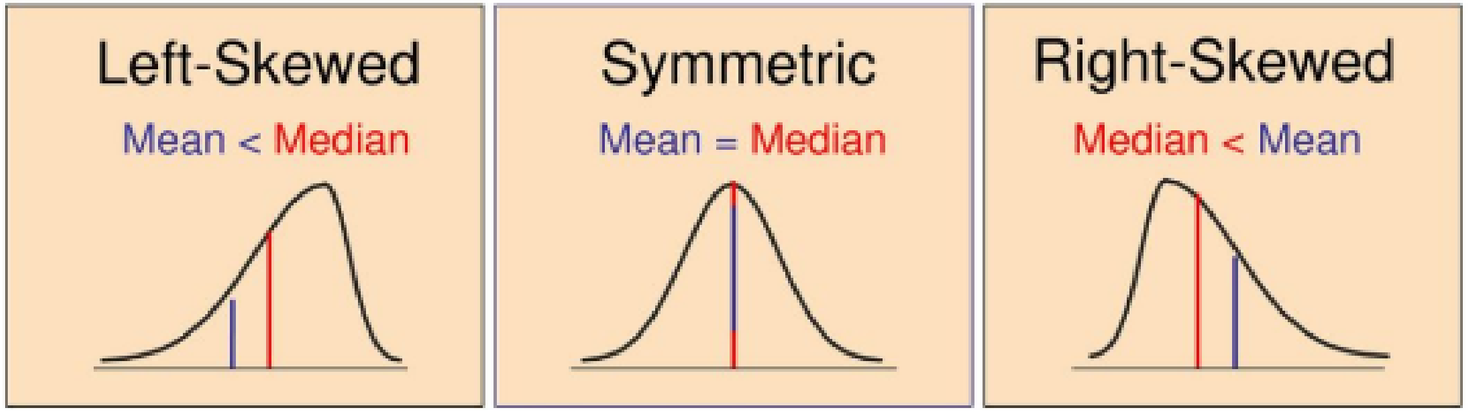
Cara Mendeteksi Outliners

2. Melihat boxplot



Cara Mendeteksi Outliners

3. Melihat bentuk distribusi data melalui histogram. Melihat nilai skewness atau derajat ketimpangan distribusi data



Left-Skewed
Mean < Median < Mode
Skewness Negatif (-)

Symmetric
Mean = Median = Mode
Skewness mendekati 0

Right-Skewed
Mean > Median > Mode
Skewness Positif (+)

Cara Mendeteksi Outliners

4. Metode Z-score

➜ Metode Z-score dapat digunakan untuk menghitung seberapa jauh setiap data dari rata-rata dengan membagi selisih antara data dan rata-rata dengan standar deviasi. Jika nilai Z-score lebih besar dari 3 atau lebih kecil dari -3, maka data dianggap sebagai outlier.

4. Metode Interquartile range (IQR)

➜ Metode IQR adalah perbedaan antara kuartil atas dan kuartil bawah dalam satu set data. Data dianggap sebagai outlier jika jauh dari nilai kuartil atas atau kuartil bawah lebih dari 1,5 kali IQR.

Cara Mengatasi Outliners

- **Identifikasi Outlier**

➜ Hal ini dapat dilakukan dengan
cara visualisasi data menggunakan boxplot atau histogram, serta penggunaan metode statistik seperti z-score atau metode quartile range.

- **Menghapus Outliers**

➜ Salah satu cara untuk mengatasi outlier adalah dengan
menghapus data yang dianggap sebagai outlier. Namun, penghapusan data harus dilakukan dengan hati-hati dan dengan mempertimbangkan dampaknya terhadap analisis data. Jika jumlah outlier tidak signifikan, maka penghapusan dapat dilakukan.

Cara Mengatasi Outliners

- **Transformasi Data**

➜ Transformasi data dapat dilakukan untuk mengatasi outlier, misalnya dengan transformasi logaritmik atau transformasi kuadratik. Transformasi data dapat membantu untuk mengurangi dampak dari outlier pada analisis data.

- **Imputasi Data**

➜ Jika outlier dianggap sebagai data yang valid dan tidak boleh dihapus, maka imputasi data dapat dilakukan. Imputasi data adalah proses mengganti nilai outlier dengan nilai yang dianggap wajar atau masuk akal berdasarkan kriteria tertentu.

Cara Mengatasi Outliners

- **Gunakan Metode Statistik yang Tahan Outlier:**

➜ Metode statistik yang tahan
outlier, seperti median dan interquartile range, dapat digunakan untuk mengurangi dampak dari outlier pada analisis data. Metode ini lebih tahan terhadap outlier daripada metode statistik lainnya seperti mean dan standar deviasi.

IQR = Q3 - Q1

batas bawah = Q1 - 1.5 * IQR

batas atas = Q3 - 1.5 * IQR

Filtering Data

filtering adalah proses selektif dalam memilih, menyaring, atau memodifikasi dataset. Umumnya, hal ini digunakan untuk memenuhi tujuan analisis atau pemodelan tertentu

Filtering Data digunakan untuk memisahkan data relevan dan penting dengan yang tidak memenuhi kebutuhan. Beberapa kriteria atau aturan *filtering* ini mencakup:

- **Nilai numerik yang dianggap valid atau relevan.**
- **Kategori atau label tertentu yang diperlukan untuk analisis.**
- **Pola atau kondisi spesifik yang harus dipenuhi oleh data.**

Secara keseluruhan, tujuan *filtering* adalah untuk meningkatkan kualitas dan efektivitas data yang mau dianalisis atau dimodelkan.

Feature Selection

Feature Selection adalah proses mengurangi jumlah fitur atau variabel input dengan memilih fitur-fitur yang dianggap paling relevan terhadap model.

Beberapa manfaat atau tujuan feature selection adalah :

1. **Mengurangi Overfitting**
2. **Semakin sedikit redundansi dalam fitur yang digunakan, berarti mengurangi kemungkinan munculnya noise dalam model**
3. **Meningkatkan Akurasi**
4. **Meminimalkan fitur dan data yang kurang tepat atau misleading**
5. **Mengurangi beban dan waktu training**
6. **Lebih sedikit data berarti mengurangi kompleksitas algoritma dan mempercepat waktu training**

Selamat Praktikum gess~