Nama : Muhammad Reesa Rosyid

Program : Python for Data science

# Pandas

## Definisi Pandas

Pandas adalah paket Python open source yang paling sering dipakai untuk menganalisis data serta membangun sebuah machine learning. Pandas dibuat berdasarkan satu package lain bernama Numpy, yang mendukung arrays multi dimensi.

## Inisiasi Pandas

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

## Membaca file csv

In [2]:
```
df = pd.read_csv('nbaallelo.csv')
```

In [3]:
```
# Melihat besar data
df.shape

#(baris,kolom)
```

Out[3]:
```
(126314, 23)
```

In [4]:
```
# Melihat baris data
len(df)
```

Out[4]:
```
126314
```

In [5]:
```
# Lihat isi data dari atas
df.head()
```

Out[5]:

| | gameorder | game_id | lg_id | _iscopy | year_id | date_game | seasongame | is_playoffs | team_id | fran_id |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 194611010TRH | NBA | 0 | 1947 | 11/1/1946 | 1 | 0 | TRH | Huskies |
| **1** | 1 | 194611010TRH | NBA | 1 | 1947 | 11/1/1946 | 1 | 0 | NYK | Knicks |
| **2** | 2 | 194611020CHS | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | CHS | Stags |
| **3** | 2 | 194611020CHS | NBA | 1 | 1947 | 11/2/1946 | 2 | 0 | NYK | Knicks |
| **4** | 3 | 194611020DTF | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | DTF | Falcons |

5 rows × 23 columns

In [6]:
```
# Lihat isi data 10 teratas
df.head(10)
```

Out[6]:

| | gameorder | game_id | lg_id | _iscopy | year_id | date_game | seasongame | is_playoffs | team_id | fran |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 194611010TRH | NBA | 0 | 1947 | 11/1/1946 | 1 | 0 | TRH | Husk |
| **1** | 1 | 194611010TRH | NBA | 1 | 1947 | 11/1/1946 | 1 | 0 | NYK | Kni |
| **2** | 2 | 194611020CHS | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | CHS | St |
| **3** | 2 | 194611020CHS | NBA | 1 | 1947 | 11/2/1946 | 2 | 0 | NYK | Kni |
| **4** | 3 | 194611020DTF | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | DTF | Falc |
| **5** | 3 | 194611020DTF | NBA | 1 | 1947 | 11/2/1946 | 1 | 0 | WSC | Capi |
| **6** | 4 | 194611020PRO | NBA | 1 | 1947 | 11/2/1946 | 1 | 0 | BOS | Cel |
| **7** | 4 | 194611020PRO | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | PRO | Steamrol |
| **8** | 5 | 194611020STB | NBA | 1 | 1947 | 11/2/1946 | 1 | 0 | PIT | Ironn |
| **9** | 5 | 194611020STB | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | STB | Bomb |

10 rows × 23 columns

In [7]:
```
# Menampilkan semua kolom
pd.set_option("display.max.columns", None)
#pd.set_option("display.max.rows", None)
df.head(10)
```

Out[7]:

| | gameorder | game_id | lg_id | _iscopy | year_id | date_game | seasongame | is_playoffs | team_id | fran |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 194611010TRH | NBA | 0 | 1947 | 11/1/1946 | 1 | 0 | TRH | Husk |
| **1** | 1 | 194611010TRH | NBA | 1 | 1947 | 11/1/1946 | 1 | 0 | NYK | Kni |
| **2** | 2 | 194611020CHS | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | CHS | St |
| **3** | 2 | 194611020CHS | NBA | 1 | 1947 | 11/2/1946 | 2 | 0 | NYK | Kni |
| **4** | 3 | 194611020DTF | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | DTF | Falc |
| **5** | 3 | 194611020DTF | NBA | 1 | 1947 | 11/2/1946 | 1 | 0 | WSC | Capi |
| **6** | 4 | 194611020PRO | NBA | 1 | 1947 | 11/2/1946 | 1 | 0 | BOS | Cel |
| **7** | 4 | 194611020PRO | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | PRO | Steamrol |
| **8** | 5 | 194611020STB | NBA | 1 | 1947 | 11/2/1946 | 1 | 0 | PIT | Ironn |
| **9** | 5 | 194611020STB | NBA | 0 | 1947 | 11/2/1946 | 1 | 0 | STB | Bomb |

In [8]:
```
# Lihat isi data 10 terbawah
df.tail(10)
```

Out[8]:

| | gameorder | game_id | lg_id | _iscopy | year_id | date_game | seasongame | is_playoffs | team_id | f |
|---|---|---|---|---|---|---|---|---|---|---|
| **126304** | 63153 | 201506070GSW | NBA | 1 | 2015 | 6/7/2015 | 98 | 1 | CLE | Ca |
| **126305** | 63153 | 201506070GSW | NBA | 0 | 2015 | 6/7/2015 | 99 | 1 | GSW | W |
| **126306** | 63154 | 201506090CLE | NBA | 1 | 2015 | 6/9/2015 | 100 | 1 | GSW | W |
| **126307** | 63154 | 201506090CLE | NBA | 0 | 2015 | 6/9/2015 | 99 | 1 | CLE | Ca |
| **126308** | 63155 | 201506110CLE | NBA | 1 | 2015 | 6/11/2015 | 101 | 1 | GSW | W |
| **126309** | 63155 | 201506110CLE | NBA | 0 | 2015 | 6/11/2015 | 100 | 1 | CLE | Ca |
| **126310** | 63156 | 201506140GSW | NBA | 0 | 2015 | 6/14/2015 | 102 | 1 | GSW | W |
| **126311** | 63156 | 201506140GSW | NBA | 1 | 2015 | 6/14/2015 | 101 | 1 | CLE | Ca |
| **126312** | 63157 | 201506170CLE | NBA | 0 | 2015 | 6/16/2015 | 102 | 1 | CLE | Ca |
| **126313** | 63157 | 201506170CLE | NBA | 1 | 2015 | 6/16/2015 | 103 | 1 | GSW | W |

In [9]:
```
# Melihat info tipe data setiap kolom

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126314 entries, 0 to 126313
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   gameorder      126314 non-null  int64  
 1   game_id        126314 non-null  object 
 2   lg_id          126314 non-null  object 
 3   _iscopy        126314 non-null  int64  
 4   year_id        126314 non-null  int64  
 5   date_game      126314 non-null  object 
 6   seasongame     126314 non-null  int64  
 7   is_playoffs    126314 non-null  int64  
 8   team_id        126314 non-null  object 
 9   fran_id        126314 non-null  object 
 10  pts            126314 non-null  int64  
 11  elo_i          126314 non-null  float64
 12  elo_n          126314 non-null  float64
 13  win_equiv      126314 non-null  float64
 14  opp_id         126314 non-null  object 
 15  opp_fran       126314 non-null  object 
 16  opp_pts        126314 non-null  int64  
 17  opp_elo_i      126314 non-null  float64
 18  opp_elo_n      126314 non-null  float64
 19  game_location  126314 non-null  object 
 20  game_result    126314 non-null  object 
 21  forecast       126314 non-null  float64
 22  notes          5424 non-null    object 
dtypes: float64(6), int64(7), object(10)
memory usage: 22.2+ MB
```

In [10]:
```
# Statistik deskriptif sederhana
df.describe()
```

Out[10]:

| | gameorder | _iscopy | year_id | seasongame | is_playoffs | pts | el |
|---|---|---|---|---|---|---|---|
| **count** | 126314.000000 | 126314.000000 | 126314.000000 | 126314.000000 | 126314.000000 | 126314.000000 | 126314.000( |
| **mean** | 31579.000000 | 0.500000 | 1988.200374 | 43.533733 | 0.063857 | 102.729982 | 1495.236( |
| **std** | 18231.927643 | 0.500002 | 17.582309 | 25.375178 | 0.244499 | 14.814845 | 112.139 |
| **min** | 1.000000 | 0.000000 | 1947.000000 | 1.000000 | 0.000000 | 0.000000 | 1091.644 |
| **25%** | 15790.000000 | 0.000000 | 1975.000000 | 22.000000 | 0.000000 | 93.000000 | 1417.237 |
| **50%** | 31579.000000 | 0.500000 | 1990.000000 | 43.000000 | 0.000000 | 103.000000 | 1500.945 |
| **75%** | 47368.000000 | 1.000000 | 2003.000000 | 65.000000 | 0.000000 | 112.000000 | 1576.060( |
| **max** | 63157.000000 | 1.000000 | 2015.000000 | 108.000000 | 1.000000 | 186.000000 | 1853.104 |

# Eksplorasi Data [EDA]

In [11]:
```
# Lihat team_id sering muncul berserta jumlah
# df['team_id'].value_counts
df.team_id.value_counts
```

Out[11]:
```
<bound method IndexOpsMixin.value_counts of 0         TRH
1         NYK
2         CHS
3         NYK
4         DTF
         ...
126309    CLE
126310    GSW
126311    CLE
126312    CLE
126313    GSW
Name: team_id, Length: 126314, dtype: object>
```

In [12]:
```
# save to file
#df["team_id"].value_counts().to_csv('team_id.csv')
df.team_id.value_counts().to_csv('team_id.csv')
```

In [13]:
```
df.fran_id.value_counts
```

Out[13]:
```
<bound method IndexOpsMixin.value_counts of 0           Huskies
1            Knicks
2             Stags
3            Knicks
4           Falcons
            ...
126309    Cavaliers
126310     Warriors
126311    Cavaliers
126312    Cavaliers
126313     Warriors
Name: fran_id, Length: 126314, dtype: object>
```

## Query sederhana

In [14]:
```
df.fran_id == "Lakers"
```

Out[14]:
```
0         False
1         False
2         False
3         False
4         False
          ...
126309    False
126310    False
126311    False
126312    False
126313    False
Name: fran_id, Length: 126314, dtype: bool
```

In [15]:
```
kondisi = df.fran_id == "Lakers"
df[kondisi]["team_id"]
```

Out[15]:
```
1136      MNL
1152      MNL
1159      MNL
1170      MNL
1183      MNL
         ...
126016    LAL
126052    LAL
126086    LAL
126115    LAL
126137    LAL
Name: team_id, Length: 6024, dtype: object
```

In [16]:
```
# Konfersi tipe data
df['date_game'] = pd.to_datetime(df['date_game'])
```

In [17]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126314 entries, 0 to 126313
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   gameorder      126314 non-null  int64
 1   game_id        126314 non-null  object
 2   lg_id          126314 non-null  object
 3   _iscopy        126314 non-null  int64
 4   year_id        126314 non-null  int64
 5   date_game      126314 non-null  datetime64[ns]
 6   seasongame     126314 non-null  int64
 7   is_playoffs    126314 non-null  int64
 8   team_id        126314 non-null  object
 9   fran_id        126314 non-null  object
 10  pts            126314 non-null  int64
 11  elo_i          126314 non-null  float64
 12  elo_n          126314 non-null  float64
 13  win_equiv      126314 non-null  float64
 14  opp_id         126314 non-null  object
 15  opp_fran       126314 non-null  object
 16  opp_pts        126314 non-null  int64
 17  opp_elo_i      126314 non-null  float64
 18  opp_elo_n      126314 non-null  float64
 19  game_location  126314 non-null  object
 20  game_result    126314 non-null  object
 21  forecast       126314 non-null  float64
 22  notes          5424 non-null    object
dtypes: datetime64[ns](1), float64(6), int64(7), object(9)
memory usage: 22.2+ MB
```

In [18]:
```
#Mengambil data game yang diadakan bulan februari
kondisi = df['date_game'].dt.month==2
df[kondisi].head(10)
```

Out[18]:

| | gameorder | game_id | lg_id | _iscopy | year_id | date_game | seasongame | is_playoffs | team_id | fr |
|---|---|---|---|---|---|---|---|---|---|---|
| **390** | 196 | 194702010NYK | NBA | 0 | 1947 | 1947-02-01 | 35 | 0 | NYK | |
| **391** | 196 | 194702010NYK | NBA | 1 | 1947 | 1947-02-01 | 36 | 0 | PHW | W |
| **392** | 197 | 194702010PRO | NBA | 0 | 1947 | 1947-02-01 | 35 | 0 | PRO | Steam |
| **393** | 197 | 194702010PRO | NBA | 1 | 1947 | 1947-02-01 | 37 | 0 | PIT | Iro |
| **394** | 198 | 194702010STB | NBA | 1 | 1947 | 1947-02-01 | 35 | 0 | CLR | F |
| **395** | 198 | 194702010STB | NBA | 0 | 1947 | 1947-02-01 | 36 | 0 | STB | Bo |
| **396** | 199 | 194702020CHS | NBA | 0 | 1947 | 1947-02-02 | 37 | 0 | CHS | |
| **397** | 199 | 194702020CHS | NBA | 1 | 1947 | 1947-02-02 | 36 | 0 | TRH | H |
| **398** | 200 | 194702020CLR | NBA | 0 | 1947 | 1947-02-02 | 36 | 0 | CLR | F |
| **399** | 200 | 194702020CLR | NBA | 1 | 1947 | 1947-02-02 | 37 | 0 | WSC | C |

In [19]:
```
kondisi1 = df['date_game'].dt.month==3
kondisi2 = df['date_game'].dt.day==11
df[kondisi1 & kondisi2].head(10)
```

Out[19]:

| | gameorder | game_id | lg_id | _iscopy | year_id | date_game | seasongame | is_playoffs | team_id | fra |
|---|---|---|---|---|---|---|---|---|---|---|
| **570** | 286 | 194703110CLR | NBA | 0 | 1947 | 1947-03-11 | 51 | 0 | CLR | Re |
| **571** | 286 | 194703110CLR | NBA | 1 | 1947 | 1947-03-11 | 55 | 0 | PIT | Iron |
| **572** | 287 | 194703110GSW | NBA | 0 | 1947 | 1947-03-11 | 53 | 0 | PHW | War |
| **573** | 287 | 194703110GSW | NBA | 1 | 1947 | 1947-03-11 | 54 | 0 | WSC | Cap |
| **574** | 288 | 194703110TRH | NBA | 1 | 1947 | 1947-03-11 | 52 | 0 | STB | Bom |
| **575** | 288 | 194703110TRH | NBA | 0 | 1947 | 1947-03-11 | 53 | 0 | TRH | Hus |
| **1050** | 526 | 194803110BLB | NBA | 0 | 1948 | 1948-03-11 | 45 | 0 | BLB | Baltir |
| **1051** | 526 | 194803110BLB | NBA | 1 | 1948 | 1948-03-11 | 44 | 0 | CHS | S |
| **1052** | 527 | 194803110GSW | NBA | 1 | 1948 | 1948-03-11 | 44 | 0 | STB | Bom |
| **1053** | 527 | 194803110GSW | NBA | 0 | 1948 | 1948-03-11 | 45 | 0 | PHW | War |

In [20]:
```
kondisi = df['team_id'] == "MNL"
df[kondisi]['date_game'].min()
```

Out[20]:
```
Timestamp('1948-11-04 00:00:00')
```

In [21]:
```
df.loc[df["team_id"] == "MNL", "date_game"].min()
```

Out[21]:
```
Timestamp('1948-11-04 00:00:00')
```

## Agregasi

In [22]:
```
kondisi = df['team_id'] == "MNL"
df[kondisi]['date_game'].agg(("min","max"))
```

Out[22]:
```
min   1948-11-04
max   1960-03-26
Name: date_game, dtype: datetime64[ns]
```

In [23]:
```
kondisi = df['team_id'] == "MNL"
df[kondisi]['win_equiv'].agg(("min","max", "mean", "std"))
```

Out[23]:
```
min     24.525501
max     66.217308
mean    45.702914
std     10.443832
Name: win_equiv, dtype: float64
```

# Deep Dive to Pandas

## Tipe Data

In [24]:
```
data = [0, 0.25, 0.5, 0.75, 1]
np_data = np.array(data)
ser_data = pd.Series(data)
```

In [25]:
```
# List
data
```

Out[25]:
```
[0, 0.25, 0.5, 0.75, 1]
```

In [26]:
```
# Array
np_data
```

Out[26]:
```
array([0.  , 0.25, 0.5 , 0.75, 1.  ])
```

In [27]:
```
# Series
ser_data
```

Out[27]:
```
0    0.00
1    0.25
2    0.50
3    0.75
4    1.00
dtype: float64
```

In [28]:
```
ser_data.values
```

Out[28]:
```
array([0.  , 0.25, 0.5 , 0.75, 1.  ])
```

In [29]:
```
ser_data.index
```

Out[29]:
```
RangeIndex(start=0, stop=5, step=1)
```

## Data Index

In [30]:
```
ser_data_idx =pd.Series(data, index=['a','b','c','d','e'])
```

In [31]:
```
ser_data_idx
```

Out[31]:
```
a    0.00
b    0.25
c    0.50
d    0.75
e    1.00
dtype: float64
```

In [ ]:
```

```

In [32]:
```
my_dict = {'kaset':100000,
           'CD':50000,
           'DVD':200000}
ser_dict = pd.Series(my_dict)
ser_dict
```

Out[32]:
```
kaset    100000
CD        50000
DVD      200000
dtype: int64
```

## DataFrame

In [33]:
```
nilai = [[76, 80, 77, 90],
        [90, 85, 60, 70],
        [88, 89,78, 90],
        [60, 50, 70, 40]] #mat, ipa, bindo, bing
df = pd.DataFrame(nilai, columns=['matematika', 'ipa', 'bahasa_indonesia', 'bahasa_inggr
df
```

Out[33]:

| | matematika | ipa | bahasa_indonesia | bahasa_inggris |
|---|---|---|---|---|
| **Budi** | 76 | 80 | 77 | 90 |
| **Danu** | 90 | 85 | 60 | 70 |
| **Eka** | 88 | 89 | 78 | 90 |
| **Wendi** | 60 | 50 | 70 | 40 |

In [34]:
```
df['matematika']
```

Out[34]:
```
Budi     76
Danu     90
Eka      88
Wendi    60
Name: matematika, dtype: int64
```

In [35]:
```
df['matematika']['Budi']
```

Out[35]:
```
76
```

Tips : Pada dataframe yang dipanggil kolom dulu baru baris

# Perbedaan loc dan iloc

In [37]:
```
data = [0, 0.25, 0.5, 0.75, 1]
ser_data_idx =pd.Series(data, index=['a','b','c','d','e'])
ser_data_idx
```

Out[37]:
```
a    0.00
b    0.25
c    0.50
d    0.75
e    1.00
dtype: float64
```

## Loc

Memanggil data dengan explisit index

In [38]:
```
ser_data_idx.loc['c']
```

Out[38]:
```
0.5
```

In [40]:
```
# Slicing loc
ser_data_idx.loc['c':'e']
```

Out[40]:
```
c    0.50
d    0.75
e    1.00
dtype: float64
```

## Iloc

Memanggil data dengan implisit index

In [42]:
```
ser_data_idx.iloc[1]
```

Out[42]:
```
0.25
```

In [45]:
```
# Slicing Iloc
ser_data_idx.iloc[1:4]
```

Out[45]:
```
b    0.25
c    0.50
d    0.75
dtype: float64
```