# DATA MANIPULATION (PART 1)

Understanding where we are as a brand and how our competitors work.

# OUR TEAM

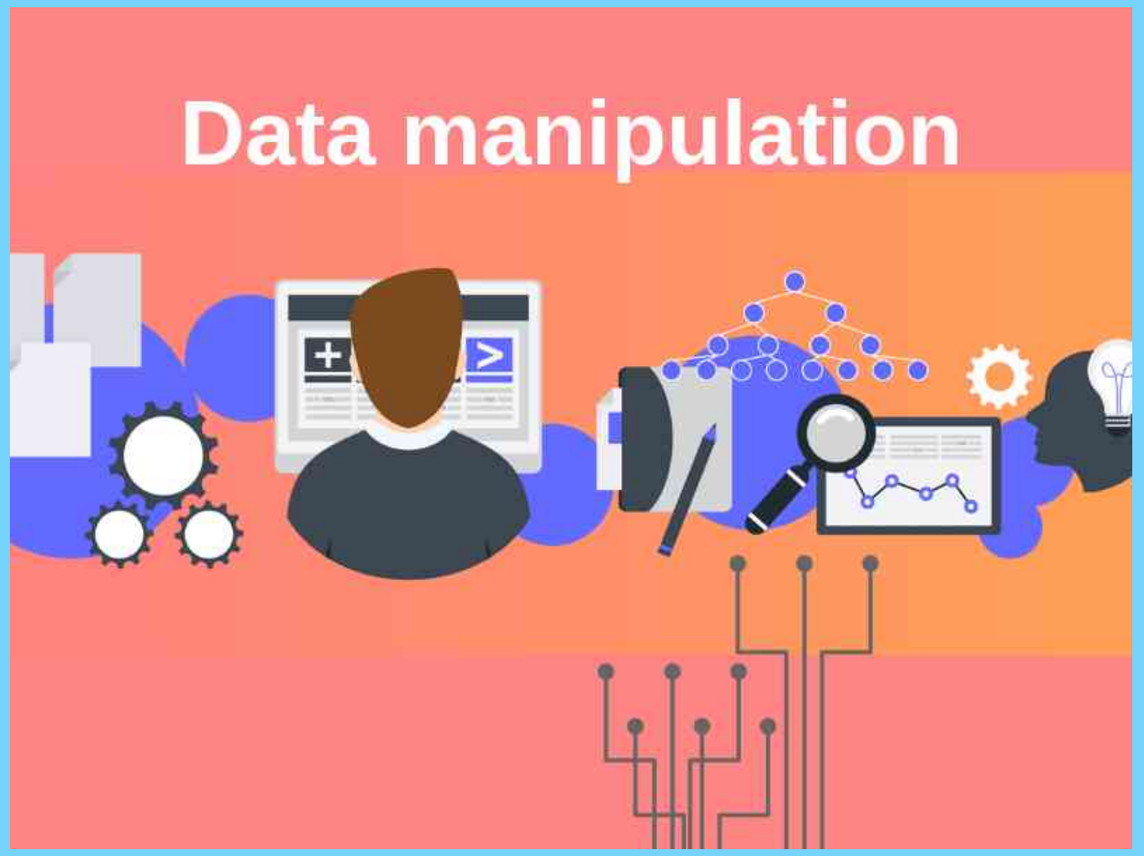
1. Andhika Utomo
2. Daril Hana Salsabila
3. Dean Setyawan
4. Heru Stiawan
5. M. Haikal Febrian P.
6. Raisya Amanah Nurwidev

# THIS WEEK MATERI

1. Data frame and series in pandas
2. index insplisit and eksplisit

# INTRODUCTION

Data manipulation is the process of changing or altering data in order to make it more readable and organized. For example, you can arrange data alphabetically to expedite the process of finding useful information. Another example of data manipulation is website management.



# SERIES IN PANDAS

A pandas Series is a one-dimensional labelled data structure which can hold data such as strings, integers and even other Python objects.

```
import pandas as pd
import numpy as np

data=[0,0.25,0.5,0.75,1]
data=pd.Series(data)
print(data)
```

```
0    0.00
1    0.25
2    0.50
3    0.75
4    1.00
dtype: float64
```

# DATAFRAME IN PANDAS

DataFrame is a 2-dimensional labeled data structure with columns of potentially different types

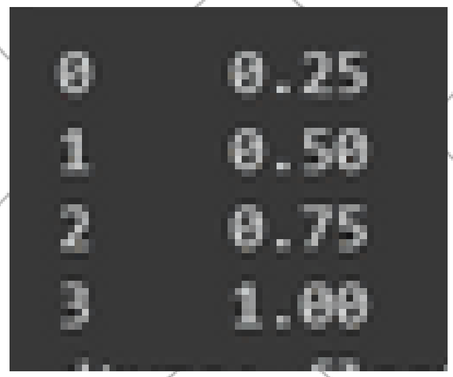
```
import pandas as pd
import numpy as np

data=[0,0.25,0.5,0.75,1]
data=pd.DataFrame(data)
print(data)
```

```
      0
0  0.00
1  0.25
2  0.50
3  0.75
4  1.00
```

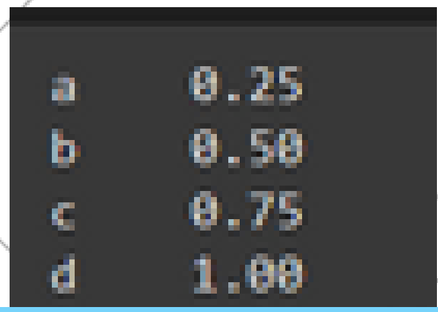
# INDEX IMPLIST AND EXPLISIT

- index implisit (null index) is an default index in python
- index explicit is an customized indexin python

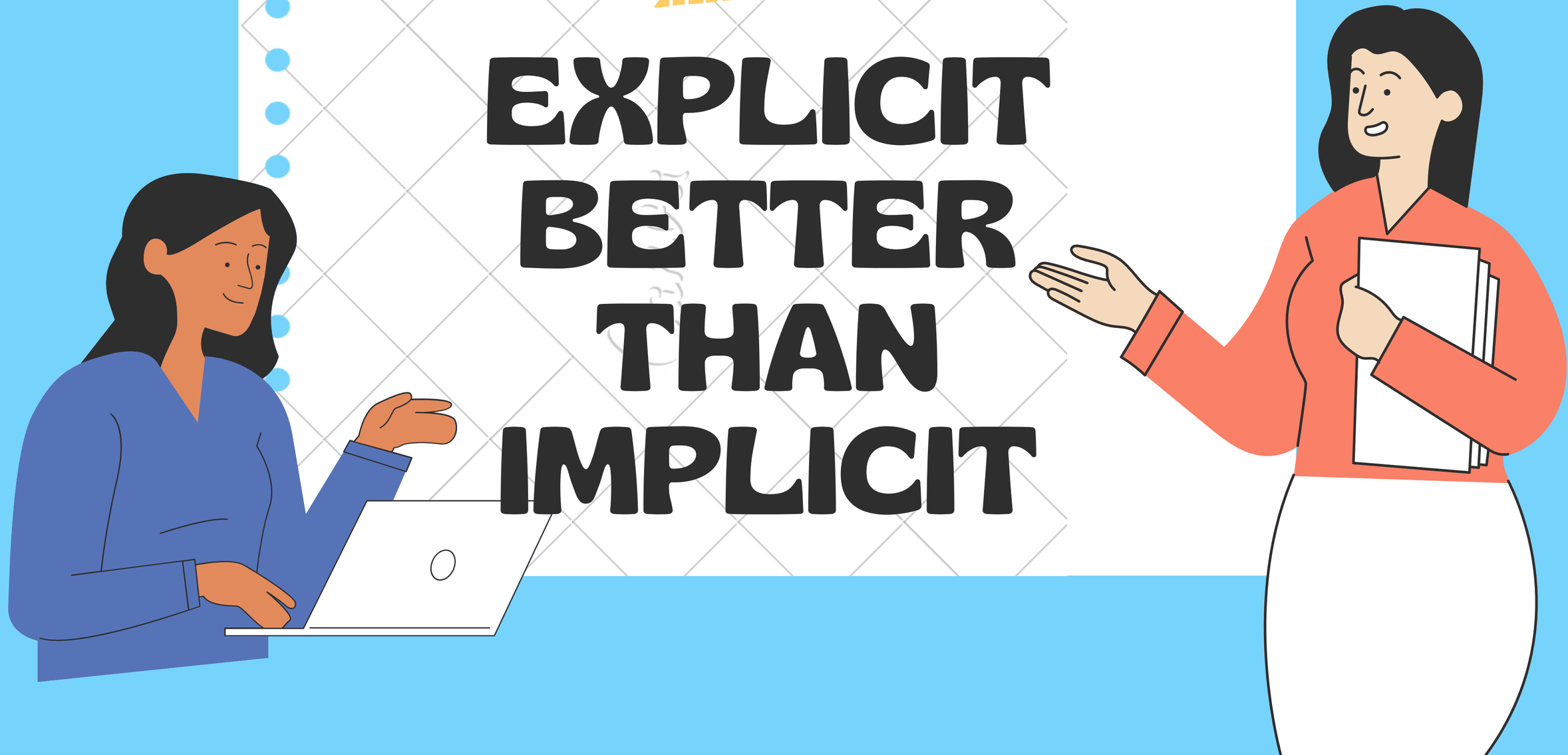
1 implicit index




2 explicit index



# EXPLICIT BETTER THAN IMPLICIT



# EXPILICT BETTER THAN IMPLICIT

Explicit is better than implicit is a sentence we heard when we learn about indexing in pandas. these sentence means if we do slicing using index. excplicit index will be prioritize than implicit index if both of index have same member of index

# ILOC AND LOC

```
data_2

2    0.25
5    0.50
3    0.75
7    1.00
dtype: float64

data_2.iloc[3]

1.0
```

iloc is to call implicit index



```
data_2

2    0.25
5    0.50
3    0.75
7    1.00
dtype: float64

data_2.loc[3]

0.75
```

ioc is to call explicit index

# WHY NEED TO USE ILOC AND LOC?

“

because iloc and loc is very useful in mainting clean and readable code

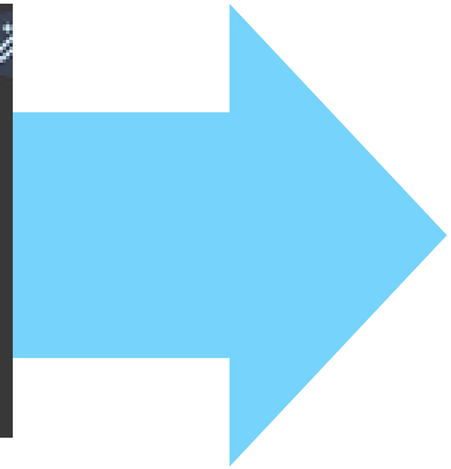
# HOW TO ADD A NEW ROW

code:

```
daerah_tambahan=pd.DataFrame({"bandung":[342,980,342/980]})
daerah_tambahan=daerah_tambahan.T
daerah_tambahan.columns=daerah.columns
pd.concat([daerah,daerah_tambahan])
```

output:

| | populasi | luas | populasi per area |
|---|---|---|---|
| jakarta | 370 | 720 | 0.513889 |
| bogor | 490 | 230 | 2.130435 |
| depok | 350 | 560 | 0.625000 |
| tanggeringan | 270 | 420 | 0.642857 |
| bekasi | 670 | 290 | 2.310345 |



| | populasi | luas | populasi per area |
|---|---|---|---|
| jakarta | 370.0 | 720.0 | 0.513889 |
| bogor | 490.0 | 230.0 | 2.130435 |
| depok | 350.0 | 560.0 | 0.625000 |
| tanggeringan | 270.0 | 420.0 | 0.642857 |
| bekasi | 670.0 | 290.0 | 2.310345 |
| bandung | 342.0 | 980.0 | 0.348980 |

# THANK YOU!

Have a great day ahead.