



Muhammad Ramadiansyah

Finaccel Machine Learning Engineer

Experiences in **Machine Learning Engineer**,
Data Scientist, **Full Stack Web Developer** and
Product Manager.

Working Experiences



Education Background



2013-2017
Bachelor Degree
Electrical
Engineering



2019-2021
Master Degree
Computer Science

Data Processing

Introduction to Pandas
DataFrame & Transformation



Topik Data Processing

Topik 1 : *Pandas Dataframe and Transformation*

1. Pengenalan DataFrame dan Pandas
2. Cara membuat DataFrame
3. Operasi sederhana DataFrame
4. Manipulasi Baris
5. Manipulasi Kolom
6. Export DataFrame ke Excel/CSV

Topik 3 : *DataFrame Combination*

1. Append (UNION) DataFrame dengan Dictionary
2. Append DataFrame dengan DataFrame
3. Merge (JOIN) DataFrame

Topik 2 : *DataFrame Aggregation*

1. GroupBy Function
2. Aggregation Function
3. Pivot Table

Pandas DataFrame & Transformation

- Pengenalan DataFrame
- Pengenalan Pandas
- Membuat DataFrame
- Operasi Dasar DataFrame
- Mengubah Tipe Data
- Filtering Column
- Membuat Column Baru
- Menghapus Column
- Menghapus Row
- Sorting Row
- Filtering Row
- Export DataFrame

Hands-On Requirement:

Hands - On :

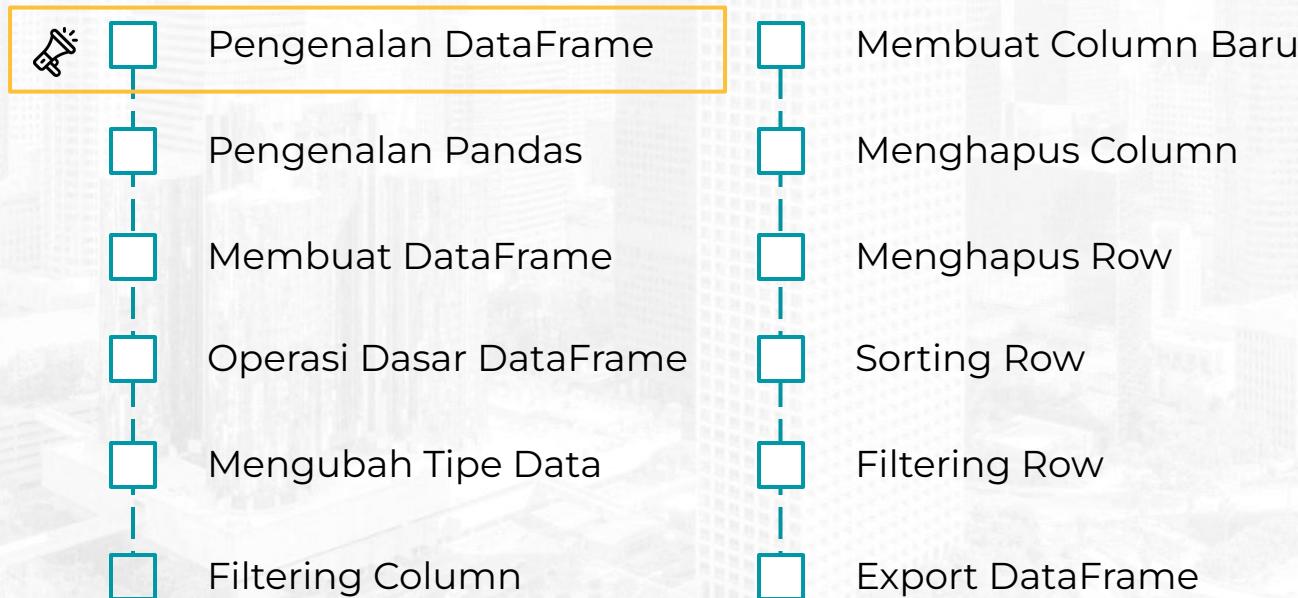
Data Processing - Pengenalan DataFrame & Pandas.ipynb

Dataset :

data_rakamin_customer.xlsx

Klik di sini
untuk akses dataset dan handsOn
Jupyter Notebook

Pandas DataFrame & Transformation



Pengenalan **DataFrame**

Apa itu *Data Frame*?

“**DataFrame** adalah **2-dimensional struktur data**, dengan kolom-kolom yang berisikan suatu informasi data .”

“**DataFrame** dapat dibayangkan seperti **spreadsheet (Excel)** atau SQL table..”

No		Nama	Alamat	Usia
0	1	Andi Gunawan	Jakarta	21
1	2	Beni Tri	Bogor	24
2	3	Cinta Laudy	Bekasi	29
3	4	Deni Hermawan	Malang	24
4	5	Endang Gusti	Aceh	23

Contoh DataFrame

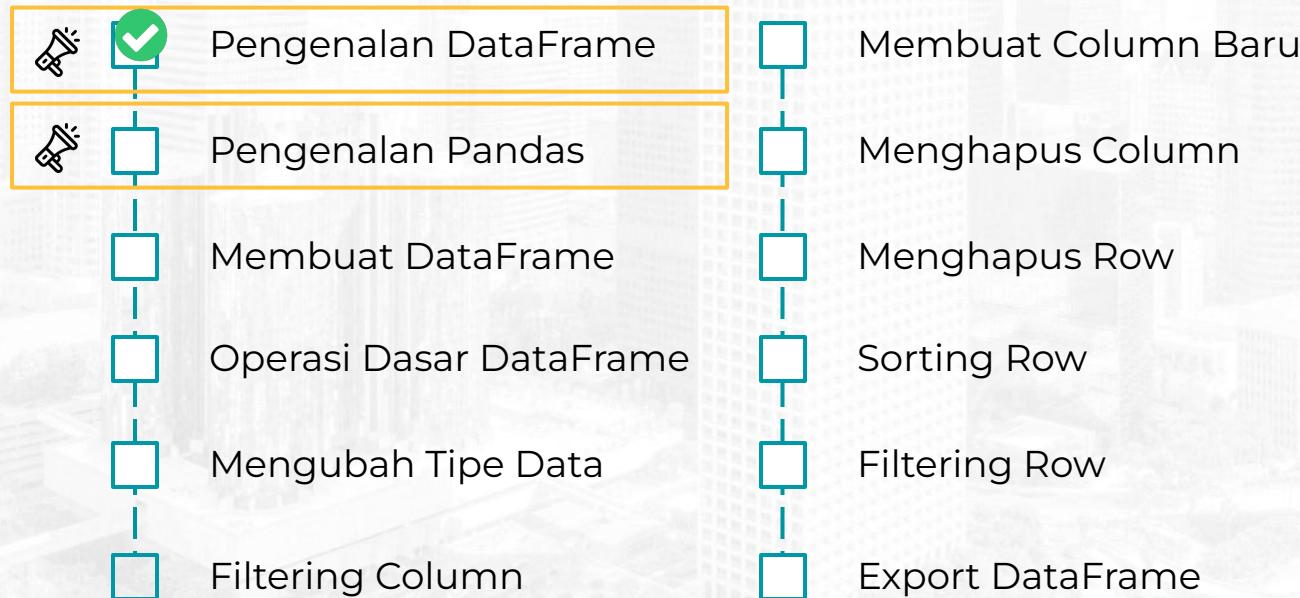
Komponen Data Frame

	No	Nama	Alamat	Usia	Columns
Index	0	1	Andi Gunawan	Jakarta	21
	1	2	Beni Tri	Bogor	24
	2	3	Cinta Laudya	Bekasi	29
	3	4	Deni Hermawan	Malang	24
	4	5	Endang Gusti	Aceh	23

Rows

Values

Pandas DataFrame & Transformation



Pengenalan **Pandas**

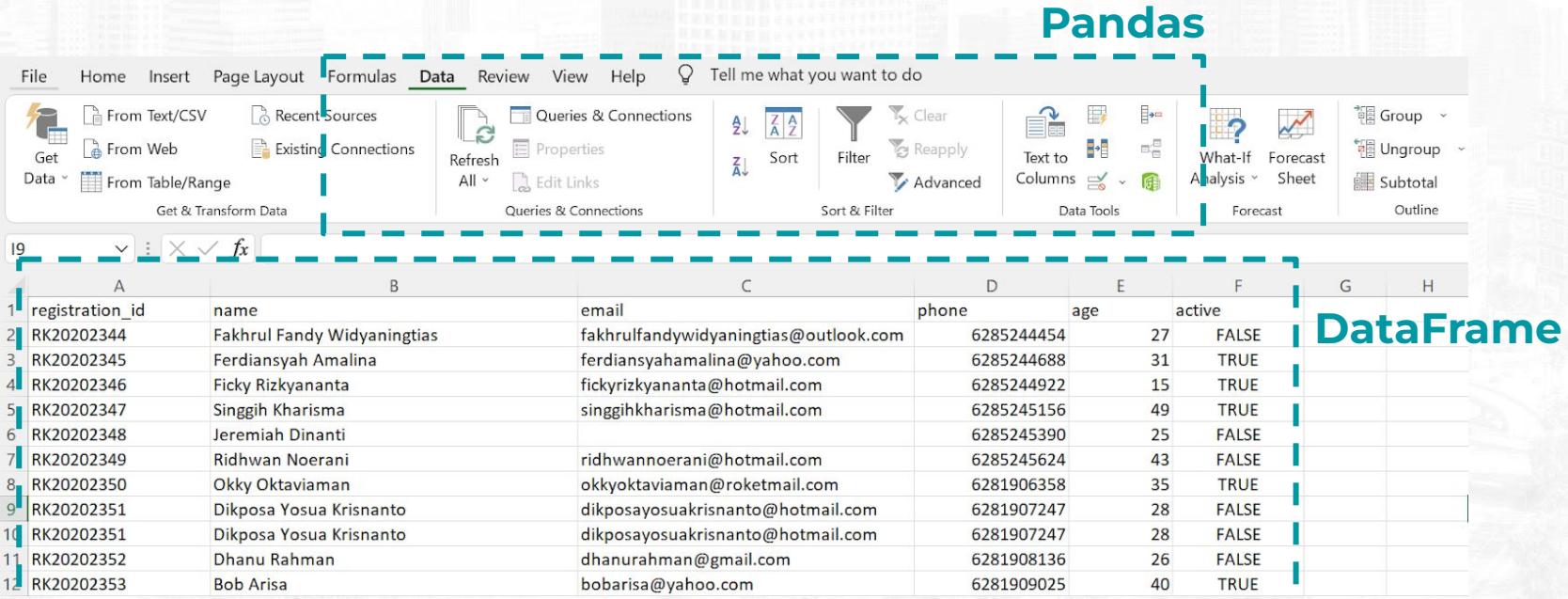
Apa itu *Pandas*?

“Pandas adalah open-source data tool
untuk manipulasi dan analisis data.”

<https://pandas.pydata.org/>

Analogi Data Frame dan Pandas

Pandas

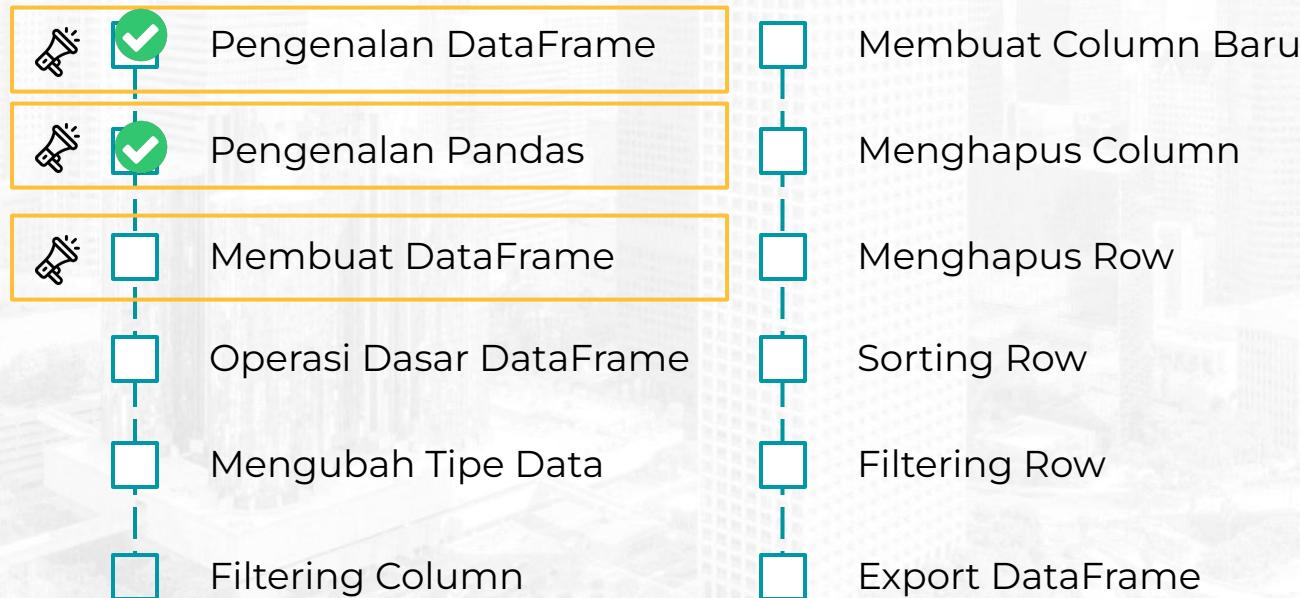


The screenshot shows a Microsoft Excel interface with the Data tab selected in the ribbon. The ribbon also includes File, Home, Insert, Page Layout, Formulas, Review, View, Help, and a search bar. Below the ribbon, there are several groups of buttons: Get & Transform Data (Get, From Text/CSV, From Web, From Table/Range), Queries & Connections (Recent Sources, Existing Connections, Refresh All, Properties, Edit Links), Sort & Filter (Sort, Filter, Advanced), Data Tools (Text to Columns, PivotTable, Subtotal), Forecast (What-If Analysis, Forecast Sheet, Forecast), and Outline (Group, Ungroup, Subtotal). The main area of the screen displays a DataFrame table with columns registration_id, name, email, phone, age, and active.

DataFrame

	A	B	C	D	E	F	G	H
1	registration_id	name	email	phone	age	active		
2	RK20202344	Fahrul Fandy Widyaningtias	fahrulfandywidyaningtias@outlook.com	6285244454	27	FALSE		
3	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6285244688	31	TRUE		
4	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6285244922	15	TRUE		
5	RK20202347	Singgih Kharisma	singgikhharisma@hotmail.com	6285245156	49	TRUE		
6	RK20202348	Jeremiah Dinanti		6285245390	25	FALSE		
7	RK20202349	Ridhwan Noerani	ridhwannoerani@hotmail.com	6285245624	43	FALSE		
8	RK20202350	Okky Oktaviaman	okkyoktaviaman@roketmail.com	6281906358	35	TRUE		
9	RK20202351	Dikposa Yosua Krisnanto	dikposayosuakrisnanto@hotmail.com	6281907247	28	FALSE		
10	RK20202351	Dikposa Yosua Krisnanto	dikposayosuakrisnanto@hotmail.com	6281907247	28	FALSE		
11	RK20202352	Dhanur Rahman	dhanurahman@gmail.com	6281908136	26	FALSE		
12	RK20202353	Bob Arisa	bobarisa@yahoo.com	6281909025	40	TRUE		

Pandas DataFrame & Transformation



Cara Membuat **DataFrame**

Secara Manual | Import data (CSV/Excel)

Membuat DataFrame Manual

Contoh Table

nama	umur	hobi	sepatu
dadan	30	memasak	nike

```
import pandas as pd
```

```
df = pd.DataFrame({ 'nama' : [ 'dadang' ],  
                    'umur' : [ 30 ],  
                    'hobi' : [ 'memasak' ],  
                    'sepatu' : [ 'nike' ]  
})
```

key menjadi nama kolom

value menjadi nilai dari tiap kolomnya

Membuat DataFrame Manual

Contoh Table

nama	umur	hobi	sepatu
dadan	30	memasak	nike
didin	40	berkebun	adidas

dataframe 2 baris

```
df = pd.DataFrame( { 'nama' : [ 'dadang' , 'didin' ] ,  
                     'umur' : [ 30 , 40] ,  
                     'hobi' : [ 'memasak' , 'berkebun' ] ,  
                     'sepatu' : [ 'nike' , 'adidas' ]  
                   } )
```

Membuat DataFrame Manual

Contoh Table

nama	umur	hobi	sepatu
dadan	30	memasak	nike
didin	40	berkebun	adidas
dodon	26	basket	puma

dataframe 3 baris

```
df = pd.DataFrame( { 'nama' : [ 'dadon' , 'didin' , 'dodon' ] ,  
                     'umur' : [ 30 , 40 , 26 ] ,  
                     'hobi' : [ 'memasak' , 'berkebun' , 'basket' ] ,  
                     'sepatu' : [ 'nike' , 'adidas' , 'puma' ]  
                   } )
```

Membuat DataFrame Manual

Contoh Table

nama	umur	hobi	sepatu	kota
dadan	30	memasak	nike	jakarta
didin	40	berkebun	adidas	bogor
dodon	26	basket	puma	depok

dataframe
3 baris & 5 kolom

```
df = pd.DataFrame( { 'nama' : [ 'dadon' , 'didin' , 'dodon' ] ,  
                     'umur' : [ 30 , 40 , 26 ] ,  
                     'hobi' : [ 'memasak' , 'berkebun' , 'basket' ] ,  
                     'sepatu' : [ 'nike' , 'adidas' , 'puma' ] ,  
                     'kota' : [ 'jakarta' , 'bogor' , 'depok' ]  
                   } )
```

Membuat DataFrame Manual

Contoh Table

** Jika ingin **terdapat NULL** value pada kolom “**umur**”

nama	umur	hobi	sepatu	kota
dadan	30	memasak	nike	jakarta
didin	40	berkebun	adidas	bogor
dodon	NaN	basket	puma	depok

```
df = pd.DataFrame( { 'nama' : [ 'dadon' , 'didin' , 'dodon' ] ,  
                     'umur' : [ 30 , 40 , None ] ,  
                     'hobi' : [ 'memasak' , 'berkebun' , 'basket' ] ,  
                     'sepatu' : [ 'nike' , 'adidas' , 'puma' ] ,  
                     'kota' : [ 'jakarta' , 'bogor' , 'depok' ]  
                   } )
```

Membuat DataFrame Manual

Contoh Table

** Jika ingin **terdapat NULL** value pada kolom “**umur**” dan “**sepatu**”

nama	umur	hobi	sepatu	kota
dadan	30	memasak	None	jakarta
didin	40	berkebun	adidas	bogor
dodon	Nan	basket	puma	depok

```
df = pd.DataFrame( { 'nama' : [ 'dadon' , 'didin' , 'dodon' ] ,  
                     'umur' : [ 30 , 40 , None ] ,  
                     'hobi' : [ 'memasak' , 'berkebun' , 'basket' ] ,  
                     'sepatu' : [ None , 'adidas' , 'puma' ] ,  
                     'kota' : [ 'jakarta' , 'bogor' , 'depok' ]  
                   } )
```

Perbedaan NULL dan “<string kosong>” Syntax

NULL berarti tidak memiliki data

```
pd.DataFrame({'No'      : [1, 2, 3, 4, 5, 6], # ada 6 element
              'Nama'    : ['Andi Gunawan', 'Beni Tri', 'Cinta Laudya',
                           'Deni Hermawan', 'Endang Gusti', 'Firman'],
              'Alamat'  : ['Jakarta', 'Bogor', 'Bekasi', 'Bekasi', 'Aceh', None],
              'Usia'    : [21, 24, 29, 24, 23, None]
            })
```

“<string kosong>” berarti ada data, namun berupa string tanpa nilai

```
pd.DataFrame({'No'      : [1, 2, 3, 4, 5, 6], # ada 6 element
              'Nama'    : ['Andi Gunawan', 'Beni Tri', 'Cinta Laudya',
                           'Deni Hermawan', 'Endang Gusti', 'Firman'],
              'Alamat'  : ['Jakarta', 'Bogor', 'Bekasi', 'Bekasi', 'Aceh', ''],
              'Usia'    : [21, 24, 29, 24, 23, None]
            })
```

Perbedaan NULL dan “<string kosong>” Tampilan

NULL berarti tidak memiliki data

No		Nama	Alamat	Usia
0	1	Andi Gunawan	Jakarta	21.0
1	2	Beni Tri	Bogor	24.0
2	3	Cinta Laudya	Bekasi	29.0
3	4	Deni Hermawan	Bekasi	24.0
4	5	Endang Gusti	Aceh	23.0
5	6	Firman	None	NaN

None : Tampilan column untuk String NULL

NaN : Tampilan column untuk Integer NULL

“<string kosong>” berarti ada data, namun berupa string kosong (tanpa nilai)

No		Nama	Alamat	Usia
0	1	Andi Gunawan	Jakarta	21.0
1	2	Beni Tri	Bogor	24.0
2	3	Cinta Laudya	Bekasi	29.0
3	4	Deni Hermawan	Bekasi	24.0
4	5	Endang Gusti	Aceh	23.0
5	6	Firman		NaN

<string kosong>: Tampilan column untuk String kosong “”

YUK LATIHAN !!!

Membuat DataFrame

Import Data

Contoh kasus sehari-hari



data customer
diterima lewat email
dari tim operasional



data di-download ke format
excel/csv ke local disk



python™

data diolah di
Python

Membuat DataFrame

Import Data

Contoh kasus sehari-hari



data customer
langsung diambil dari
database



data di-download ke format
excel/csv ke local disk

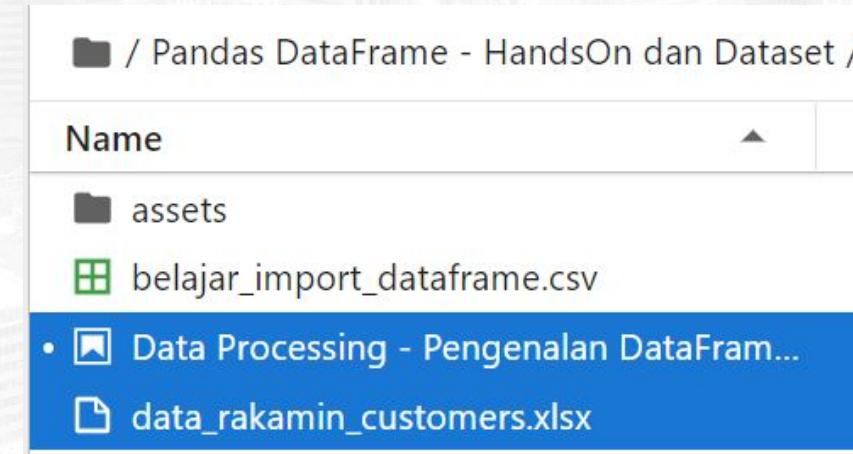


data diolah di
Python

Membuat DataFrame

Import Data

Kasus Pertama: Excel dan ipynb di Folder yang sama



- Excel :
`pd.read_excel('nama_file.xls')`
- CSV : `pd.read_csv('nama_file.csv')`

Membuat DataFrame

Import Data

Kasus Pertama: Excel dan ipynb di Folder yang sama

```
# import dan simpan dengan nama df_customers
df_customers = pd.read_excel('data_rakamin_customers.xlsx')
```

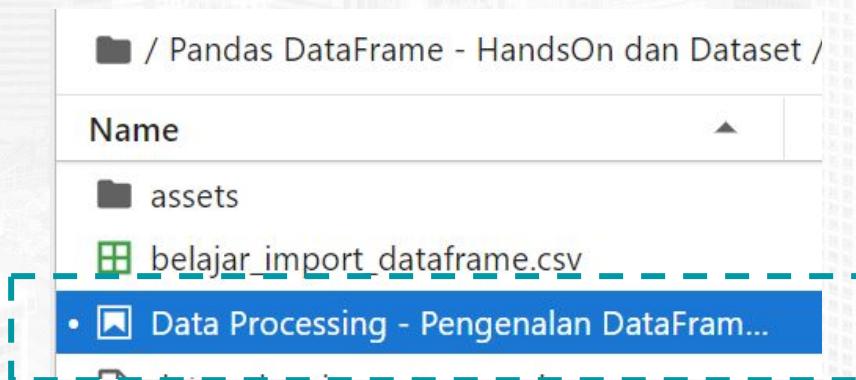
```
# run df_customers untuk melihat dataset
df_customers
```

	registration_id	name	email	phone	age	active
0	RK20202344	Fahrul Fandy Widyaningtias	fahrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False
1	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True
3	RK20202347	Singgih Kharisma	singgikhkharisma@hotmail.com	6.285245e+09	49.0	True
4	RK20202348	Jeremiah Dinanti	NaN	6.285245e+09	25.0	False
...

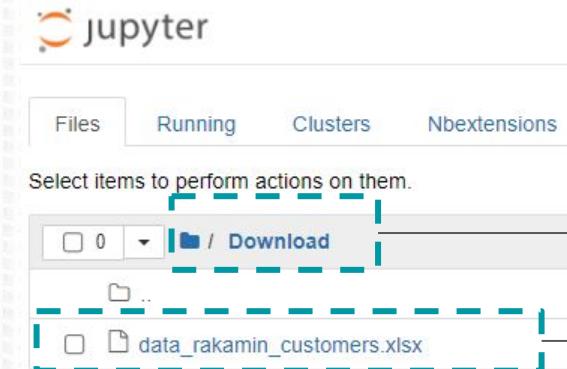
Membuat DataFrame

Import Data

Kasus Kedua: Excel dan ipynb di **Folder yang berbeda**



Lokasi python ipynb berada di folder Pandas DataFrame - HandsOn dan Dataset



Lokasi Excel berada di folder Download

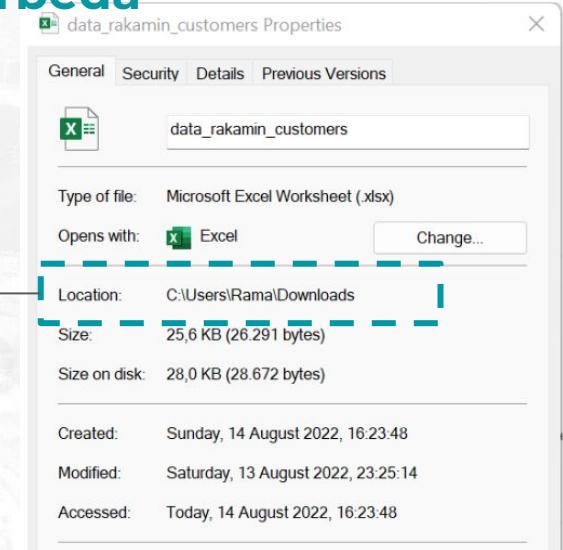
Membuat DataFrame

Import Data

Kasus Kedua: Excel dan ipynb di **Folder yang berbeda**

- Excel : pd.read_excel('PATH/nama_file.xlsx')
- CSV : pd.read_csv('PATH/nama_file.csv')

- cek lokasi dari excel/csv
- klik kanan pada file
- klik properties
- copy-paste location
- ubah backslash \ menjadi slash /



Membuat DataFrame

Import Data

Kasus Kedua: Excel dan ipynb di **Folder yang berbeda**

```
# Import menggunakan Excel file di folder yang berbeda
df_customers = pd.read_excel(r'C:\Users\Rama\Downloads\data_rakamin_customers.xlsx')
```

```
# run df_customers untuk melihat dataset
df_customers.head()
```

	registration_id	name	email	phone	age	income_group	active	city	province	hobby	...	website_
0	RK20202344	Fakhru Fandy Widyaningtias	fakhru_fandy_widyaningtias@outlook.com	6.285244e+09	27.0		2	False	Tanjungpinang	Kepulauan Riau	Amateur radio	...
1	RK20202345	Ferdiansyah Amalina	ferdiansyah_amalina@yahoo.com	6.285245e+09	31.0		3	True	Bogor	Jawa Barat	Animation	...
2	RK20202346	Ficky Rizkyananta	ficky_rizkyananta@hotmail.com	6.285245e+09	15.0		1	True	Kota Administrasi Jakarta Pusat	Jakarta	Anime	...
3	RK20202347	Singgih Kharisma	singgih_kharisma@hotmail.com	6.285245e+09	49.0		4	True	Metro	Lampung	Aquascaping	...

Kesalahan yang sering terjadi saat Import Data

- **Salah syntax**

Data Excel di folder yang berbeda,
tetapi menggunakan syntax khusus di folder yang sama

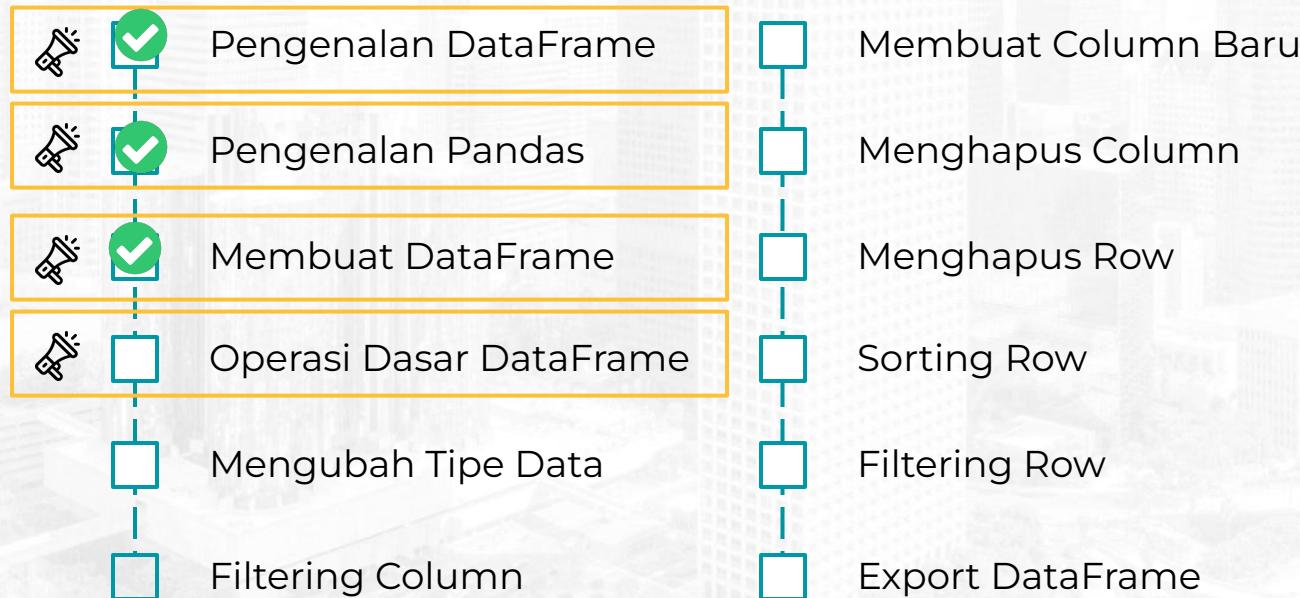
- **Typo**

Typo yang sering terjadi:

- `pd.read_csv('data_rakamin_customers-customers.csv')` ← nama dataset salah
- `pd.read_csv(data_rakamin_customer.csv)` ← Tidak menggunakan kutip “ ”
- `pd.read_csv('data_rakamin_customer').csv` ← ekstensi setelah kurung tutup)
- `pd.read_csv('data_rakamin_customers.xlsx')` ← data berupa xlsx tetapi pakai read_csv

YUK LATIHAN !!!

Pandas DataFrame & Transformation



Operasi Dasar **DataFrame**

.shape | .info() | .describe() | .head() | .tail()

Operasi Dasar

.shape

melihat **Jumlah Baris dan Kolom** dari dataframe

```
# Format df.shape
```

```
df_customers.shape  
  
(107, 21)
```

```
#mengambil hanya total baris dari data
```

```
df_customers.shape[0]  
  
107
```

```
#mengambil hanya total kolom dari data
```

```
df_customers.shape[1]  
  
21
```

Operasi Dasar

.info()

melihat **informasi dasar** dari dataframe

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107 entries, 0 to 106
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   registration_id    107 non-null    object  
 1   name                107 non-null    object  
 2   email               107 non-null    object  
 3   phone               104 non-null    float64 
 4   age                 106 non-null    float64 
 5   income_group        107 non-null    int64  
 6   active              107 non-null    bool   
 7   city                107 non-null    object  
 8   province            107 non-null    object  
 9   hobby               102 non-null    object  
 10  fav_movie           69 non-null    object  
 11  website_visit_count 107 non-null    int64  
 12  website_recency     107 non-null    int64  
 13  website_purchase_rate 107 non-null    float64 
 14  website_purchase_amount 107 non-null    float64 
 15  website_cs_call_count 107 non-null    int64  
 16  accept_campaign_1    107 non-null    bool   
 17  accept_campaign_2    107 non-null    bool   
 18  accept_campaign_3    107 non-null    bool   
 19  accept_campaign_4    107 non-null    bool   
 20  accept_campaign_5    107 non-null    bool  
dtypes: bool(6), float64(4), int64(4), object(7)
memory usage: 13.3+ KB
```

tipe data tiap columns

- object : string
- float64 : bil real, memiliki angka decimal
- integer : bil bulat
- bool : boolean (true/false)

Operasi Dasar

.describe()

melihat **statistik sederhana** dari column bertipe integer/float

```
# Syntax Format df.describe()  
  
df_customers.describe()
```

	phone	age
count	1.030000e+02	105.000000
mean	6.121534e+09	33.190476
std	9.560833e+08	11.520129
min	6.284456e+08	15.000000
25%	6.285239e+09	24.000000
50%	6.286801e+09	33.000000
75%	6.287881e+09	44.000000
max	6.289633e+09	57.000000

percentile data :

- 25% : Percentile 25 tiap kolom
- 50% : Percentile 50 tiap kolom (median)
- 75% : Percentile 75 tiap kolom

Operasi Dasar

.head()

melihat **n baris pertama**, paling sering digunakan untuk skimming data...

```
# Syntax Format df.head(N), default N=5
df_customers.head()
```

	registration_id	name		email	phone	age	active
0	RK20202344	Fakhru	L Fandy Widyaningtias	fakhru.fandy.widyaningtias@outlook.com	6.285244e+09	27.0	False
1	RK20202345	Ferdiansyah	Amalina	ferdiansyah.amalina@yahoo.com	6.285245e+09	31.0	True
2	RK20202346	Ficky	Rizkyananta	ficky.rizkyananta@hotmail.com	6.285245e+09	15.0	True
3	RK20202347	Singgih	Kharisma	singgih.kharisma@hotmail.com	6.285245e+09	49.0	True
4	RK20202348	Jeremiah	Dinanti	NaN	6.285245e+09	25.0	False

Operasi Dasar

.head()

melihat **n baris pertama**, paling sering digunakan untuk skimming data...

Contoh Lain

3 Baris pertama dari data

```
df_customers.head(3)
```

	registration_id	name	email	phone	age	active
0	RK20202344	Fahrul Fandy Widyaningtias	fahrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False
1	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True

Operasi Dasar

.head()

melihat **n baris pertama**, paling sering digunakan untuk skimming data...

Contoh Lain

7 Baris pertama dari data

```
df_customers.head(7)
```

	registration_id	name		email	phone	age	active
0	RK20202344	Fakhru	Fandy Widyaningtias	fakhru_fandy_widyaningtias@outlook.com	6.285244e+09	27.0	False
1	RK20202345	Ferdiansyah	Amalina	ferdiansyah_amalina@yahoo.com	6.285245e+09	31.0	True
2	RK20202346	Ficky	Rizkyananta	ficky_rizkyananta@hotmail.com	6.285245e+09	15.0	True
3	RK20202347	Singgih	Kharisma	singgih_kharisma@hotmail.com	6.285245e+09	49.0	True
4	RK20202348	Jeremiah	Dinanti		NaN	25.0	False
5	RK20202349	Ridhwan	Noerani	ridhwannoerani@hotmail.com	6.285246e+09	43.0	False
6	RK20202350	Okky	Oktaviaman	okky_oktaviaman@roketmail.com	6.281906e+09	35.0	True

Operasi Dasar

.tail()

melihat **n baris terakhir**, paling sering digunakan untuk skimming data...

```
# Format df.tail(N), default N=5  
  
df_customers.tail()
```

	registration_id	name	email	phone	age	active
101	RK20202439	Fajar Gusti	fajargusti@hotmail.com	6.283155e+09	54.0	True
102	RK20202440	Amalia Azizah Putriana	amaliaazizahputriana@yahoo.com	6.283542e+09	57.0	True
103	RK20202441	Didin Suherdin	didinsuherdin@yahoo.com	6.281082e+09	44.0	True
104	RK20202442	Amelia Sari Putri	ameliasariputri@ymail.com	6.284594e+09	32.0	True
105	RK20202443	Puspita Amalia	puspitaamalia@hotmail.com	6.288690e+09	27.0	True

Operasi Dasar

.tail()

melihat **n baris terakhir**, paling sering digunakan untuk skimming data...

Contoh Lain

4 Baris terakhir dari data

```
df_customers.tail(4)
```

	registration_id	name	email	phone	age	active
102	RK20202440	Amalia Azizah Putriana	amaliaazizahputriana@yahoo.com	6.283542e+09	57.0	True
103	RK20202441	Didin Suherdin	didinsuherdin@yahoo.com	6.281082e+09	44.0	True
104	RK20202442	Amelia Sari Putri	ameliasariputri@ymail.com	6.284594e+09	32.0	True
105	RK20202443	Puspita Amalia	puspitaamalia@hotmail.com	6.288690e+09	27.0	True

Operasi Dasar

.tail()

melihat **n baris terakhir**, paling sering digunakan untuk skimming data...

Contoh Lain

6 Baris terakhir dari data

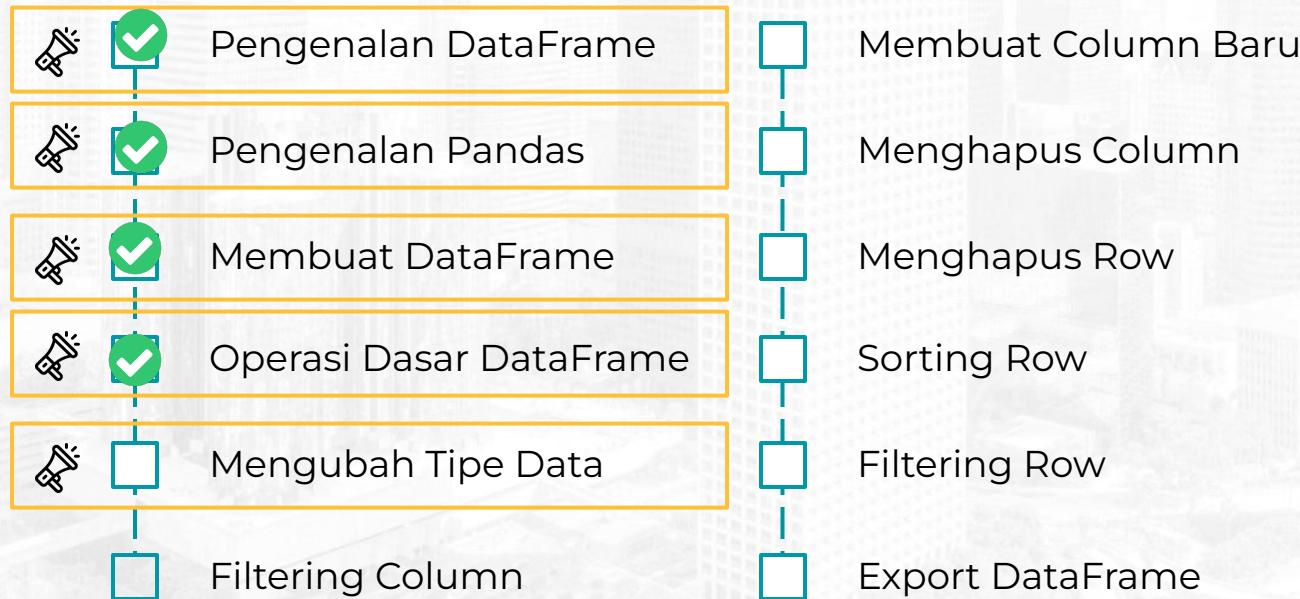
```
df_customers.tail(6)
```

	registration_id	name	email	phone	age	active
100	RK20202438	Nurul Latifa	nurullatifa@ymail.com	6.288888e+09	34.0	True
101	RK20202439	Fajar Gusti	fajargusti@hotmail.com	6.283155e+09	54.0	True
102	RK20202440	Amalia Azizah Putriana	amaliaazizahputriana@yahoo.com	6.283542e+09	57.0	True
103	RK20202441	Didin Suherdin	didinsuherdin@yahoo.com	6.281082e+09	44.0	True
104	RK20202442	Amelia Sari Putri	ameliasariputri@ymail.com	6.284594e+09	32.0	True
105	RK20202443	Puspita Amalia	puspitaamalia@hotmail.com	6.288690e+09	27.0	True

YUK LATIHAN !!!

Manipulasi column **DataFrame**

Pandas DataFrame & Transformation



Mengubah Tipe Data **DataFrame**

Ubah Tipe Data

.astype()

Mengubah tipe data dari Integer ke String atau sebaliknya

```
# Check dahulu tipe data masing-masing kolom
df_customers.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106 entries, 0 to 105
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   registration_id  106 non-null    object  
 1   name              106 non-null    object  
 2   email             106 non-null    object  
 3   phone             103 non-null    float64 
 4   age               105 non-null    float64 
 5   active            106 non-null    bool    
 6   city              106 non-null    object  
 7   province          106 non-null    object  
dtypes: bool(1), float64(2), object(5)
memory usage: 6.0+ KB
```

Kita akan coba ubah kolom **Phone** ke

- String, dan
- Integer

Ubah Tipe Data

.astype()

Format Syntax

```
df['column'] = df['column'].astype(tipe_data)
```

tipe_data bisa di-input satu dari List berikut

str : Jika ingin mengubah ke string

int : Jika ingin mengubah ke Integer

float : Jika ingin mengubah ke float

Ubah Tipe Data

.astype()

Contoh

Ubah kolom **phone** menjadi **String**

```
# Ubah kolom phone ke string
df_customers['phone'] = df_customers['phone'].astype(str)
```

```
# Cek hasil menggunakan .info()
df_customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106 entries, 0 to 105
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   registration_id  106 non-null    object 
 1   name              106 non-null    object 
 2   email             106 non-null    object 
 3   phone             106 non-null    object 
 4   age               105 non-null    float64
 5   active            106 non-null    bool  
 6   city              106 non-null    object 
 7   province          106 non-null    object 
dtypes: bool(1), float64(1), object(6)
memory usage: 6.0+ KB
```

Kolom Phone

sudah menjadi Object / String

Ubah Tipe Data

.astype()

Contoh

Ubah kolom **phone** menjadi **Float**

```
# Ubah kolom phone (sebelumnya string) ke float
df_customers['phone'] = df_customers['phone'].astype(float)

# Cek hasil menggunakan .info()
df_customers.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106 entries, 0 to 105
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
---  --  
 0   registration_id  106 non-null    object 
 1   name              106 non-null    object 
 2   email             106 non-null    object 
 3   phone             103 non-null    float64
 4   age               105 non-null    float64
 5   active            106 non-null    bool   
 6   city              106 non-null    object 
 7   province          106 non-null    object 
dtypes: bool(1), float64(2), object(5)
memory usage: 6.0+ KB
```

Kolom Phone
sudah menjadi Float

Ubah Tipe Data .astype()

Contoh

Ubah kolom **phone** menjadi **Integer**

```
# Ubah kolom phone (sebelumnya float) ke integer
df_customers['phone'] = df_customers['phone'].astype(int)
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-31-5c63af883c5e> in <module>
      1 # Ubah kolom phone (sebelumnya float) ke integer
----> 2 df_customers['phone'] = df_customers['phone'].astype(int)

~\anaconda3\lib\site-packages\pandas\core\generic.py in astype(self, dtype, copy, errors)
    5875         else:
    5876             # else, only a single dtype is given
-> 5877             new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
    5878             return self._constructor(new_data).__finalize__(self, method="astype")
    5879
-----
```

ValueError: Cannot convert non-finite values (NA or inf) to integer

Error message: ada baris dimana kolom **Phone** bernilai NA atau NULL

Ubah Tipe Data

.astype()

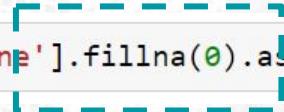
Contoh

Ubah kolom **phone** menjadi **Integer (SOLUTION)**

Beberapa solusi yang bisa diambil:

1. Jika kolom sudah bertipe float, maka **tidak perlu convert** ke Integer karena sifat float dan Integer mirip saat melakukan analisis
2. Gunakan syntax **.fillna(0) untuk mengisi NULL** value menjadi nilai 0.

```
df_customers['phone'] = df_customers['phone'].fillna(0).astype(int)
```



Data NULL akan berubah menjadi 0,
setelahnya convert to Integer

Ubah Tipe Data

.str.lower()

Mengubah kolom **menjadi huruf kecil semua**

Format Syntax

```
df['column'] = df['column'].str.lower()
```

Contoh

Ubah kolom **name** menjadi huruf kecil semua

```
df_customers['name'] = df_customers['name'].str.lower()
```

```
df_customers.head() #Cek hasilnya
```

	registration_id	name	email	phone	age	active
0	RK20202344	fakhrul fandy widyaningtias	fakhrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False
1	RK20202345	ferdiansyah amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True
2	RK20202346	ficky rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True
3	RK20202347	singgih kharisma	singgikhkharisma@hotmail.com	6.285245e+09	49.0	True
4	RK20202348	jeremiah dinanti	jeremia@rakamin.com	6.285245e+09	25.0	False

Ubah Tipe Data

.str.upper()

Mengubah kolom **menjadi huruf BESAR semua**

Format Syntax

```
df['column'] = df['column'].str.upper()
```

Contoh

Ubah kolom **name** menjadi huruf BESAR semua

```
df_customers['name'] = df_customers['name'].str.upper()
```

```
df_customers.head() #Cek hasilnya
```

	registration_id	name	email	phone	age	active
0	RK20202344	FAKHRUL FANDY WIDYANINGTIAS	fakhrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False
1	RK20202345	FERDIANSYAH AMALINA	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True
2	RK20202346	FICKY RIZKYANANTA	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True
3	RK20202347	SINGGIH KHASRIMA	singgihkharisma@hotmail.com	6.285245e+09	49.0	True
4	RK20202348	JEREMIAH DINANTI	jeremia@rakamin.com	6.285245e+09	25.0	False

Ubah Tipe Data

Recap

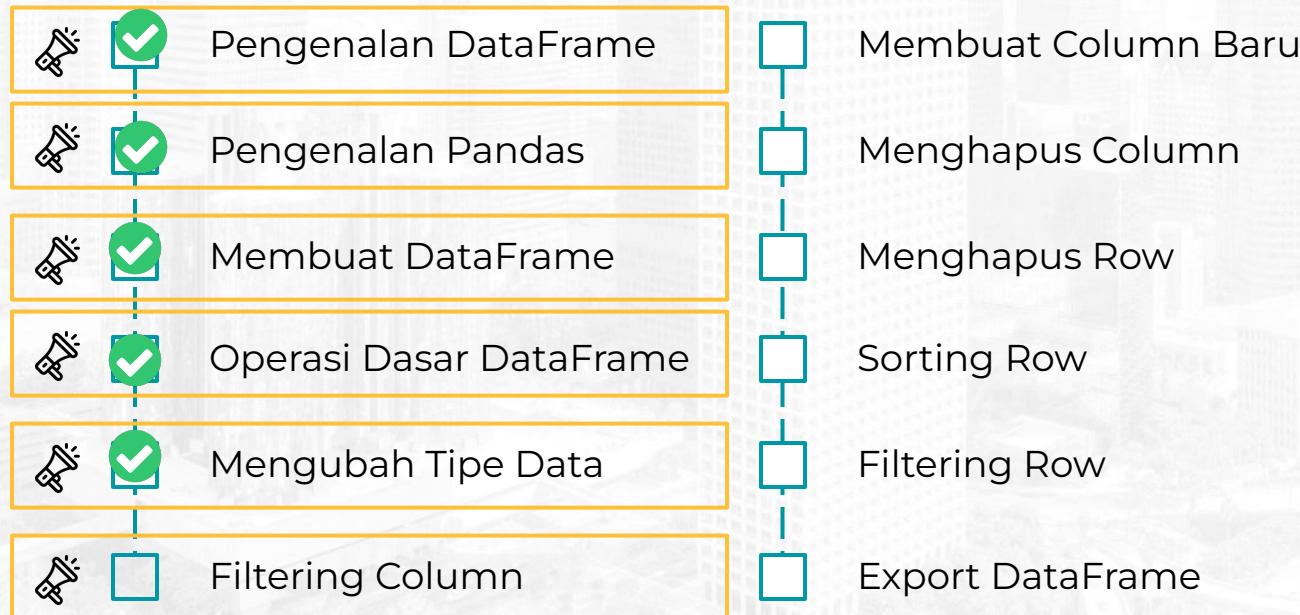
Syntax	Kegunaan	Syarat
<code>df['column'].astype(int)</code>	Ubah tipe data ke Integer	Tidak ada baris NULL
<code>df['column'].astype(float)</code>	Ubah tipe data ke Float	-
<code>df['column'].astype(str)</code>	Ubah tipe data ke String	-
<code>df['column'].str.lower()</code>	Ubah menjadi huruf kecil	-
<code>df['column'].str.upper()</code>	Ubah menjadi huruf BESAR	-

Mau lebih lengkap? [Klik ini](#)



YUK LATIHAN !!!

Pandas DataFrame & Transformation



Filtering column **DataFrame**

- Tanpa filter, kolom terdisplay akan banyak dan memanjang ke kanan
- Pada beberapa kasus, kita hanya butuh beberapa kolom untuk dilihat

df_customers.head()												
	registration_id	name	email	phone	age	income_group	active	city	province	hobby	...	webs
0	RK20202344	Fahrul Fandy Widyaningtias	fahrulfandywidyaningtias@outlook.com	6.285244e+09	27.0		2	False	Tanjungpinang	Kepulauan Riau	Amateur radio	...
1	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0		3	True	Bogor	Jawa Barat	Animation	...
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0		1	True	Kota Administrasi Jakarta Pusat	Jakarta	Anime	...
3	RK20202347	Singgih Kharisma	singgihkharisma@hotmail.com	6.285245e+09	49.0		4	True	Metro	Lampung	Aquascaping	...
4	RK20202348	Jeremiah Dinanti	jeremia@rakamin.com	6.285245e+09	25.0		2	False	Malang	Jawa Timur	Art	...

5 rows × 21 columns

Column Filtering

Format Syntax

`df[column_list]`

Kita list kolom yang dibutuhkan

Column Filtering

Contoh

Filter data supaya tersedia kolom name, email dan phone saja

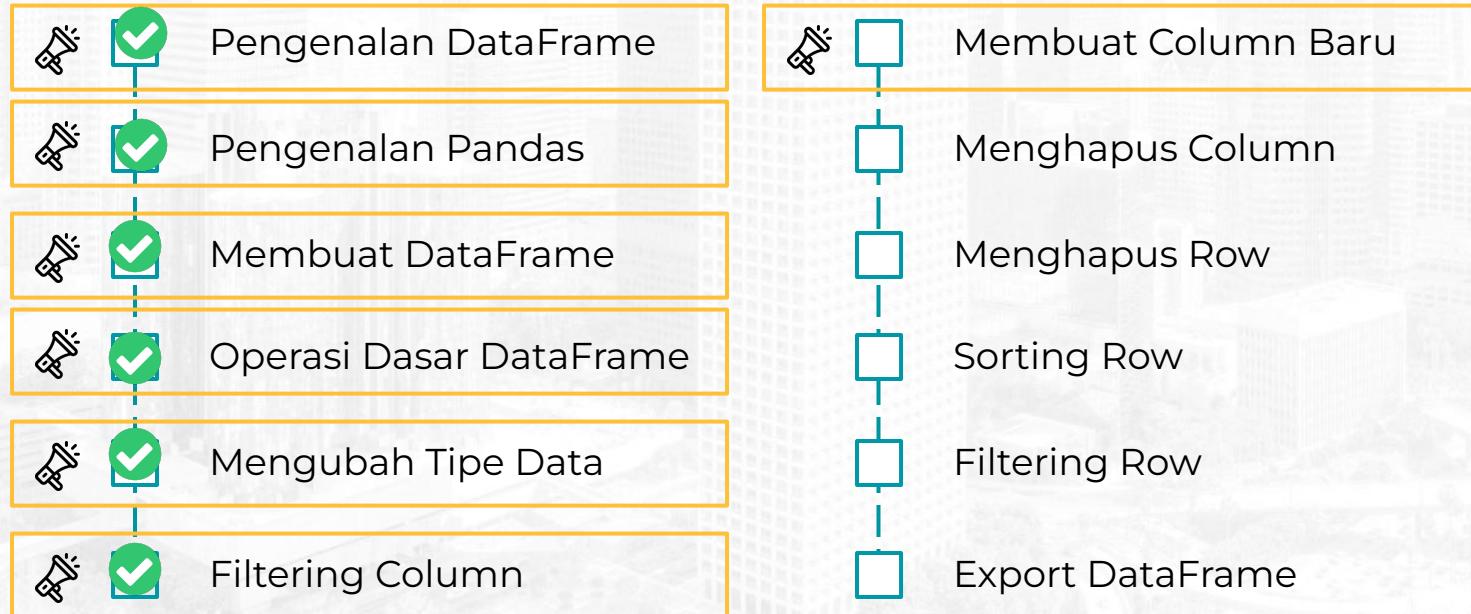
```
df_customers[['name','email','phone']]
```

	name	email	phone
0	Fakhru Fandy Widyaningtias	fakhru fandywidyaningtias@outlook.com	6.285244e+09
1	Ferdiansyah Amalina	ferdiansyah amalina@yahoo.com	6.285245e+09
2	Ficky Rizkyananta	ficky rizkyananta@hotmail.com	6.285245e+09
3	Singgih Kharisma	singgih kharisma@hotmail.com	6.285245e+09
4	Jeremiah Dinanti	jeremia@rakamin.com	6.285245e+09

YUK LATIHAN !!!

Outline Pembelajaran

Pandas DataFrame & Transformation



Menambah **Kolom Baru**

dari sebuah List | kolom lain

Menambah Kolom

Dari sebuah List

Format Syntax

df['new_column_name'] = List

No		Nama	Alamat	Usia	no_sepatu
0	1	Andi Gunawan	Jakarta	21	
1	2	Beni Tri	Bogor	24	
2	3	Cinta Laudya	Bekasi	29	
3	4	Deni Hermawan	Malang	24	
4	5	Endang Gusti	Aceh	23	

dataframe

Cara:

- Buat List `no_sepatu` dahulu
- Kemudian gunakan format syntax di atas

Menambah Kolom

Dari sebuah List

```
# Buat List berisikan nomor sepatu
no_sepatu = [ 40, 41, 42, 39, 42]

# Tambahkan no_sepatu ke dataframe
data['no_sepatu'] = no_sepatu

# cek 5 baris teratas untuk melihat apakah data sudah masuk
data.head()
```

** jumlah elemen pada List harus sama dengan jumlah baris pada dataframe

No		Nama	Alamat	Usia	no_sepatu
0	1	Andi Gunawan	Jakarta	21	40
1	2	Beni Tri	Bogor	24	41
2	3	Cinta Laudya	Bekasi	29	42
3	4	Deni Hermawan	Malang	24	39
4	5	Endang Gusti	Aceh	23	42

Menambah Kolom

Dari Kombinasi Kolom Lainnya

Format Syntax

- **perkalian** dari beberapa kolom

```
df['new_column_name'] = df['column_1'] * df['column_2']
```

- **penjumlahan** dari beberapa kolom

```
df['new_column_name'] = df['column_1'] + df['column_2']
```

- **pembagian** dari beberapa kolom

```
df['new_column_name'] = df['column_1'] / df['column_2']
```

dan kombinasi operator lainnya :) , bisa banyak use cases kok

Menambah Kolom Dari Kombinasi Kolom Lainnya

contoh 1

Buatlah kolom `tahun_kelahiran` menggunakan kolom `age` yang sudah tersedia. Asumsi: sekarang 2021

```
#Tahun kelahiran = current_year - age
df_customers['tahun_kelahiran'] = 2021 - df_customers['age']

#check hasil
df_customers
```

** Bertipe float karena age bertipe float

	registration_id	name	email	phone	age	active	city	province	tahun_kelahiran
0	RK20202344	Fakhru Fandy Widyaningtias	fakhrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False	Tanjungpinang	Kepulauan Riau	1994.0
1	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True	Bogor	Jawa Barat	1990.0
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True	Kota Administrasi Jakarta Pusat	Jakarta	2006.0

Menambah Kolom Dari Kombinasi Kolom Lainnya

contoh 2

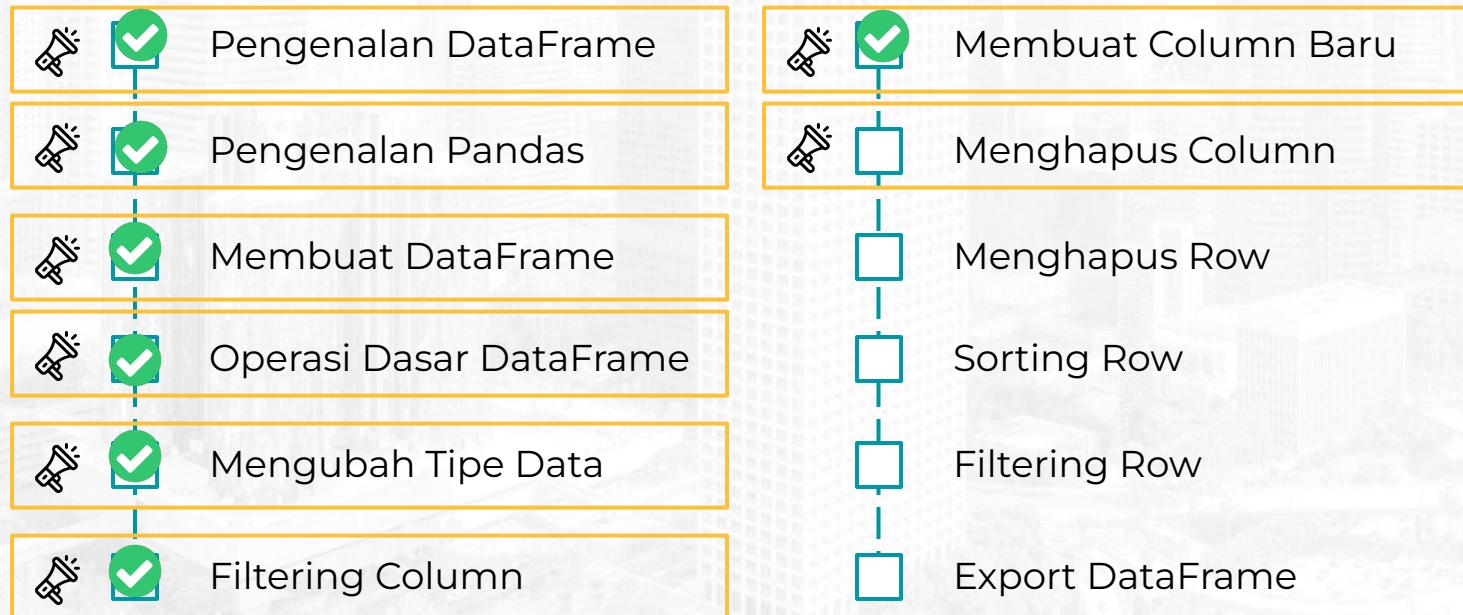
Buatlah kolom `address` dengan menggabungkan kolom `city` dan `province`

```
df_customers['address'] = df_customers['city'] + ' - ' + df_customers['province']
df_customers
```

id	name	email	phone	age	active	city	province	tahun_kelahiran	address
44	Fahrul Fandy Widyaningtias	fahrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False	Tanjungpinang	Kepulauan Riau	1994.0	Tanjungpinang - Kepulauan Riau
45	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True	Bogor	Jawa Barat	1990.0	Bogor - Jawa Barat
46	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True	Kota Administrasi Jakarta Pusat	Jakarta	2006.0	Kota Administrasi Jakarta Pusat - Jakarta
47	Singgih Kharisma	singgikhkharisma@hotmail.com	6.285245e+09	49.0	True	Metro	Lampung	1972.0	Metro - Lampung

YUK LATIHAN !!!

Pandas DataFrame & Transformation



Menghapus **Kolom**

Menghapus Satu Kolom

Format Syntax

```
df = df.drop('column_name', axis=1)
```

Menghapus Satu Kolom

contoh 1 Hapuslah kolom “**registration_id**”

```
df_customers = df_customers.drop('registration_id', axis=1)
```

	name	email	phone	age	active	city	province
0	Fakhru Fandy Widyaningtias	fakhru_fandy_widyaningtias@outlook.com	6.285244e+09	27.0	False	Tanjungpinang	Kepulauan Riau
1	Ferdiansyah Amalina	ferdiansyah_amalina@yahoo.com	6.285245e+09	31.0	True	Bogor	Jawa Barat
2	Ficky Rizkyananta	ficky_rizkyananta@hotmail.com	6.285245e+09	15.0	True	Kota Administrasi Jakarta Pusat	Jakarta
3	Singgih Kharisma	singgih_kharisma@hotmail.com	6.285245e+09	49.0	True	Metro	Lampung
4	Jeremiah Dinanti	jeremiah@rakamin.com	6.285245e+09	25.0	False	Malang	Jawa Timur

Menghapus Satu Kolom

contoh 2 Hapuslah kolom “name”

```
df_customers = df_customers.drop('name', axis=1)
```

	email	phone	age	active	city	province
0	fakhrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False	Tanjungpinang	Kepulauan Riau
1	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True	Bogor	Jawa Barat
2	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True	Kota Administrasi Jakarta Pusat	Jakarta
3	singgihkharisma@hotmail.com	6.285245e+09	49.0	True	Metro	Lampung
4	jeremia@rakamin.com	6.285245e+09	25.0	False	Malang	Jawa Timur

Menghapus Satu Kolom

contoh 3 Hapuslah kolom “city”

```
df_customers = df_customers.drop('city', axis=1)
```

	email	phone	age	active	province
0	fakhrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False	Kepulauan Riau
1	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True	Jawa Barat
2	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True	Jakarta
3	singgihkharisma@hotmail.com	6.285245e+09	49.0	True	Lampung
4	jeremia@rakamin.com	6.285245e+09	25.0	False	Jawa Timur

Menghapus Dua atau Lebih Kolom

Format Syntax

```
df = df.drop(['column_1','column_2', ... , 'column_N'], axis=1)
```

Menghapus Dua atau Lebih Kolom

contoh

Hapuslah kolom “**regitration_id**” , “**name**” , “**city**”

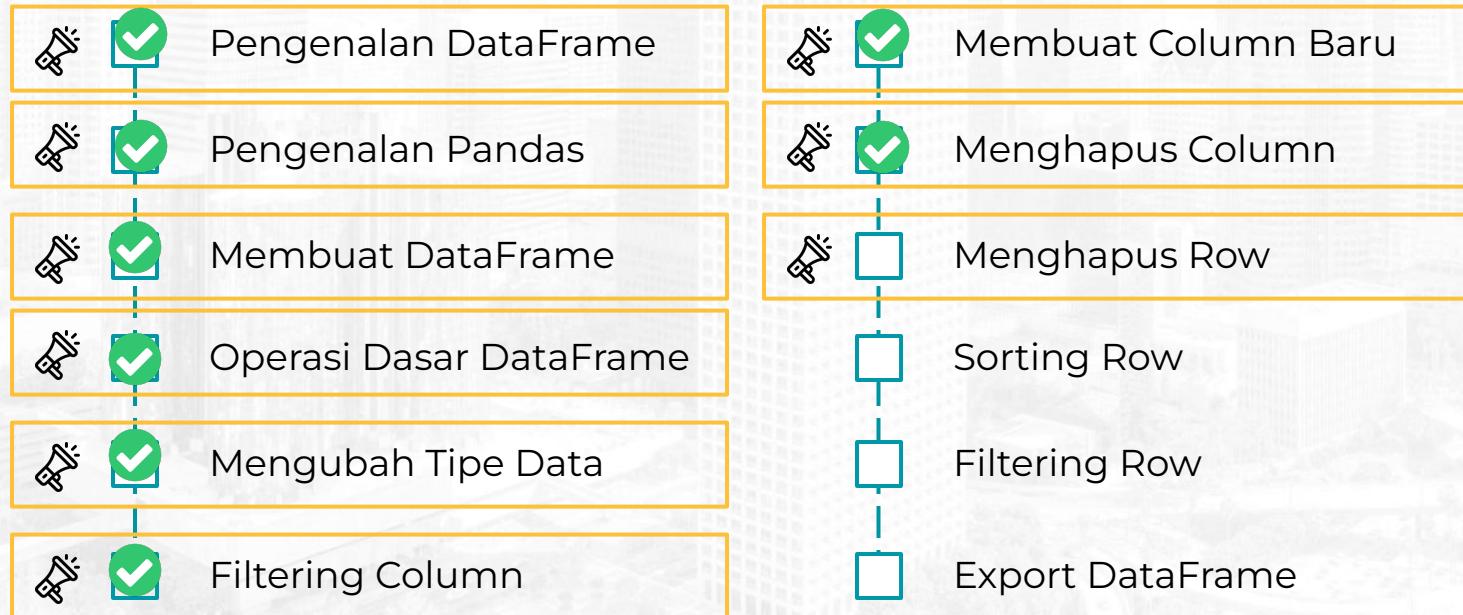
```
df_customers = df_customers.drop(['registration_id','name','city'], axis=1)
```

	email	phone	age	active	province
0	fakhrul_fandywidyaningtias@outlook.com	6.285244e+09	27.0	False	Kepulauan Riau
1	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True	Jawa Barat
2	ficky_rizkyananta@hotmail.com	6.285245e+09	15.0	True	Jakarta
3	singgih_kharisma@hotmail.com	6.285245e+09	49.0	True	Lampung
4	jeremia@rakamin.com	6.285245e+09	25.0	False	Jawa Timur

YUK LATIHAN !!!

Manipulasi row **DataFrame**

Pandas DataFrame & Transformation



Menghapus Baris Duplicate **DataFrame**

Hapus Baris Duplicate

Duplicate adalah kondisi dimana **beberapa baris memiliki nilai yang sama**

Contoh

registration_id	name	email	phone	age	active	city	province
RK20202351	Dikposa Yosua Krisnanto	dikposayosuakrisnanto@hotmail.com	6.281907e+09	28.0	False	Kota Administrasi Jakarta Pusat	Jakarta
RK20202351	Dikposa Yosua Krisnanto	dikposayosuakrisnanto@hotmail.com	6.281907e+09	28.0	False	Kota Administrasi Jakarta Pusat	Jakarta
RK20202381	Jova Imania	jovaimania@gmail.com	6.284688e+08	17.0	True	Depok	Jawa Barat
RK20202381	Jova Imania	jovaimania@gmail.com	6.284688e+08	17.0	True	Depok	Jawa Barat
RK20202383	Garin Sugiyanto	garinsugiyanto@yahoo.com	6.287881e+09	33.0	True	Purwokerto	Jawa Tengah
RK20202383	Garin Sugiyanto	garinsugiyanto@yahoo.com	6.287881e+09	33.0	True	Purwokerto	Jawa Tengah

Hapus Baris Duplicate

Format Syntax

Cek apakah ada baris yang duplicate : df.duplicated().any()

Contoh

```
df_customers.duplicated().any()
```

```
True
```

- Jika **True** maka terdapat baris yang duplicate
- Jika **False** maka tidak ada baris yang duplicate

Hapus Baris Duplicate

Format Syntax

Cek baris mana saja yang duplicate: df[df.duplicated(keep=False) == True]

Contoh

```
df_customers[df_customers.duplicated(keep=False) == True]
```

	registration_id	name	email	phone	age	active	city	province
7	RK20202351	Dikposa Yosua Krisnanto	dikposayosuakrisnanto@hotmail.com	6.281907e+09	28.0	False	Kota Administrasi Jakarta Pusat	Jakarta
8	RK20202351	Dikposa Yosua Krisnanto	dikposayosuakrisnanto@hotmail.com	6.281907e+09	28.0	False	Kota Administrasi Jakarta Pusat	Jakarta
38	RK20202381	Jova Imania	jovaimania@gmail.com	6.284688e+08	17.0	True	Depok	Jawa Barat
39	RK20202381	Jova Imania	jovaimania@gmail.com	6.284688e+08	17.0	True	Depok	Jawa Barat
41	RK20202383	Garin Sugiyanto	garinsugiyanto@yahoo.com	6.287881e+09	33.0	True	Purwokerto	Jawa Tengah
42	RK20202383	Garin Sugiyanto	garinsugiyanto@yahoo.com	6.287881e+09	33.0	True	Purwokerto	Jawa Tengah

Hapus Baris Duplicate

Format Syntax

HAPUS baris uplicate: df.drop_duplicates()

Contoh

```
# disimpan ke nama baru untuk membedakan dataframe awal dan setelah dihapus duplicate
```

```
df_customers_clean = df_customers.drop_duplicates()
```

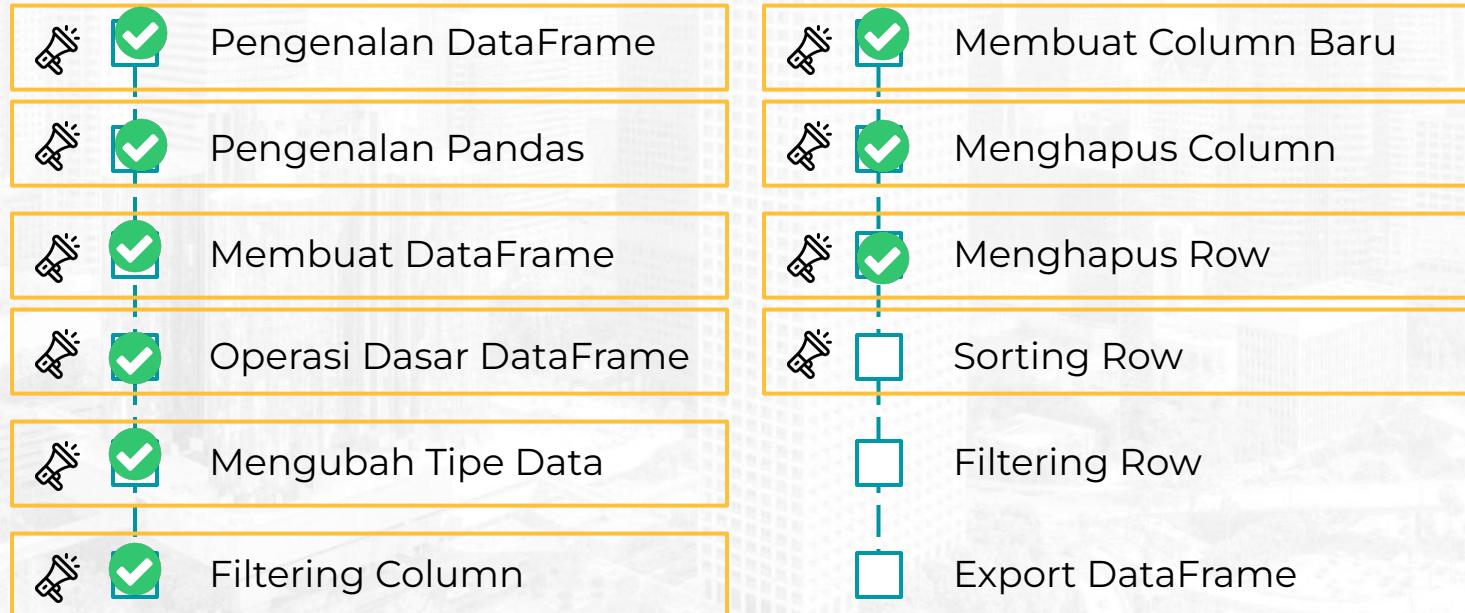
```
# Cek apakah benar sudah clean
```

```
df_customers_clean.duplicated().any()
```

False

YUK LATIHAN !!!

Pandas DataFrame & Transformation



Melakukan sorting **DataFrame**

Operasi Dasar

.sort_values()

untuk **mengurutkan** Data Frame berdasarkan kolom tertentu

Contoh

nama	umur	hobi
Dadan	60	Memasak
Dadan	15	Pingpong
Dadan	33	Renang
Heru	17	Berkebun
Andaru	26	Basket
Yonathan	55	Musik



nama	umur	hobi
Andaru	26	Basket
Dadan	15	Pingpong
Dadan	33	Renang
Dadan	60	Memasak
Heru	17	Berkebun
Yonathan	55	Musik

Before : Belum terurut

After: Terurut berdasarkan nama dan umur

Operasi Dasar

.sort_values()

untuk **mengurutkan** Data Frame berdasarkan kolom tertentu

Format Syntax

Urutkan berdasarkan **Satu kolom**

ASCENDING - Kecil → Besar

- df.sort_values('column_name', ascending = True)
- df.sort_values('column_name') default: ascending

DESCENDING : Besar → Kecil

- df.sort_values(column_name, ascending = False)

Operasi Dasar

.sort_values()

untuk **mengurutkan** Data Frame berdasarkan kolom tertentu

Contoh

Urutkan berdasarkan **Satu kolom : age**

```
df_customers.sort_values('age', ascending=True)
```

	registration_id	name	email	phone	age	active
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True
70	RK20202410	Hilary Adha	hilaryadha@yahoo.com	6.285321e+09	15.0	True
86	RK20202426	Rarahayu Verev	rarahayuverev@outlook.com	6.285216e+09	15.0	True
81	RK20202421	Ovienov Asti Idrus	ovienovastiidrus@outlook.com	6.285249e+09	15.0	True
60	RK20202400	Rianita Santi	rianitasanti@yahoo.com	6.287840e+09	15.0	True
...

Operasi Dasar

.sort_values()

untuk **mengurutkan** Data Frame berdasarkan kolom tertentu

Contoh

Urutkan berdasarkan **Satu kolom : name** (A to Z untuk string column)

```
df_customers.sort_values('name', ascending=True)
```

	registration_id	name	email	phone	age	active
35	RK20202378	Alfin Chairunisa	alfinchairunisa@gmail.com	6.284456e+08	20.0	True
64	RK20202404	Amadea Octaza	amadeaoctaza@roketmail.com	6.287814e+09	17.0	True
102	RK20202440	Amalia Azizah Putriana	amaliaazizahputriana@yahoo.com	6.283542e+09	57.0	True
67	RK20202407	Amalia Widi Abelardo	amaliawidiabelardo@outlook.com	6.285341e+09	17.0	True
80	RK20202420	Amanda Idayu	amandaidayu@yahoo.com	6.285256e+09	17.0	True
...

Operasi Dasar

.sort_values()

untuk **mengurutkan** Data Frame berdasarkan kolom tertentu

Format Syntax

Urutkan berdasarkan **Multiple kolom**

ASCENDING - Kecil → Besar

- df.sort_values(['column1', 'column2'] , ascending = [True, True])
- df.sort_values(['column1', 'column2']) → default: ascending

DESCENDING : Besar → Kecil

- df.sort_values(['column1', 'column2'] , ascending = [False, False])

Operasi Dasar

.sort_values()

untuk **mengurutkan** Data Frame berdasarkan kolom tertentu

Contoh

Urutkan berdasarkan **Multiple kolom : age & name**

```
df_customers.sort_values(['age', 'name'], ascending=[False, False])
```

	registration_id	name	email	phone	age	active
102	RK20202440	Amalia Azizah Putriana	amaliaazizahputriana@yahoo.com	6.283542e+09	57.0	True
101	RK20202439	Fajar Gusti	fajargusti@hotmail.com	6.283155e+09	54.0	True
96	RK20202434	Bagus Maulana	bagusmaulana@outlook.com	6.288953e+09	54.0	True
91	RK20202431	Prima Hermawan	primahermawan@rocketmail.com	6.281856e+09	50.0	True
92	RK20202431	Prima Hermawan	primahermawan@rocketmail.com	6.281856e+09	50.0	True
...

Operasi Dasar

.sort_values()

untuk **mengurutkan** Data Frame berdasarkan kolom tertentu

Format Syntax

Urutkan berdasarkan **Multiple kolom** - Tipe 2

ASCENDING dan DESCENDING : Urutan sesuai kebutuhan saat analisis

```
df.sort_values( [ 'column1', 'column2' ] , ascending = [True, False] )
```



** column1 secara Ascending | column2 secara Descending

Operasi Dasar

.sort_values()

untuk **mengurutkan** Data Frame berdasarkan kolom tertentu

Format Syntax

Urutkan berdasarkan **Multiple kolom** - Tipe 2

ASCENDING dan DESCENDING : Urutan sesuai kebutuhan saat analisis

```
df.sort_values( [ 'column1', 'column2' , 'column3' ] , ascending = [False, False, True] )
```



** column1 secara Descending | column2 secara Descending | column3 secara Ascending

Operasi Dasar

.sort_values()

untuk **mengurutkan** Data Frame berdasarkan kolom tertentu

Contoh

Urutkan berdasarkan **Multiple kolom : age, name dan registration_id**

```
df_customers.sort_values(['age', 'name', 'registration_id'], ascending=[False, True, True])
```

	registration_id	name	email	phone	age	active
102	RK20202440	Amalia Azizah Putriana	amaliaazizahputriana@yahoo.com	6.283542e+09	57.0	True
96	RK20202434	Bagus Maulana	bagusmaulana@outlook.com	6.288953e+09	54.0	True
101	RK20202439	Fajar Gusti	fajargusti@hotmail.com	6.283155e+09	54.0	True
44	RK20202385	Geraldi Lalo	geraldilalo@hotmail.com	6.287886e+09	50.0	True
30	RK20202373	Herdaru Junaidi	herdarujunaidi@outlook.com	6.287834e+09	50.0	True
...

YUK LATIHAN !!!

Pandas DataFrame & Transformation

- | | |
|---|--|
|   Pengenalan DataFrame |   Membuat Column Baru |
|   Pengenalan Pandas |   Menghapus Column |
|   Membuat DataFrame |   Menghapus Row |
|   Operasi Dasar DataFrame |   Sorting Row |
|   Mengubah Tipe Data |   Filtering Row |
|   Filtering Column |  Export DataFrame |

Single Filtering **DataFrame**

Basic Operator | SQL analogi LIKE , IN, Null dan Not Null

Single Filtering Basic Operator

Format Syntax

```
df[ df['column_name'] <operator> value ]
```

Operator	Kegunaan	Contoh
==	Sama dengan	age == 10
!=	Tidak sama dengan	age != 10
>	Lebih dari	age > 10
>=	Lebih dari sama dengan	age >= 10
<	Kurang dari	age < 10
<=	Kurang dari sama dengan	age <= 10

Single Filtering Basic Operator

Contoh

Filter dataset untuk **age = 15 tahun**

```
df_customers[df_customers['age']==15]
```

	registration_id	name	email	phone	age	active
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True
60	RK20202400	Rianita Santi	rianitasanti@yahoo.com	6.287840e+09	15.0	True
70	RK20202410	Hilary Adha	hilaryadha@yahoo.com	6.285321e+09	15.0	True
81	RK20202421	Ovienov Asti Idrus	ovienovastiidrus@outlook.com	6.285249e+09	15.0	True
86	RK20202426	Rarahayu Verev	rarahayuverev@outlook.com	6.285216e+09	15.0	True

Single Filtering Basic Operator

Contoh

Filter dataset untuk **age > 50 tahun**

```
df_customers[df_customers['age'] > 50]
```

	registration_id	name	email	phone	age	active
96	RK20202434	Bagus Maulana	bagusmaulana@outlook.com	6.288953e+09	54.0	True
101	RK20202439	Fajar Gusti	fajargusti@hotmail.com	6.283155e+09	54.0	True
102	RK20202440	Amalia Azizah Putriana	amaliaazizahputriana@yahoo.com	6.283542e+09	57.0	True

Single Filtering Basic Operator

Contoh

Filter dataset untuk **email = hilaryadha@yahoo.com**

```
df_customers[df_customers['email'] == 'hilaryadha@yahoo.com']
```

registration_id	name	email	phone	age	active	
70	RK20202410	Hilary Adha	hilaryadha@yahoo.com	6.285321e+09	15.0	True

Single Filtering Basic Operator

Contoh

Filter dataset untuk **nama = Rianita Santi**

```
df_customers[df_customers['name'] == 'Rianita Santi']
```

	registration_id	name	email	phone	age	active
60	RK20202400	Rianita Santi	rianitasanti@yahoo.com	6.287840e+09	15.0	True

Filter Kasus: Pertama

Gimana cara filter untuk mengambil data yang memiliki nama-nama berikut ?

- Reynard Ramadhansyah
- Putu Adinda
- Firman Reza
- Puspa Fariza
- Nadia Persadani

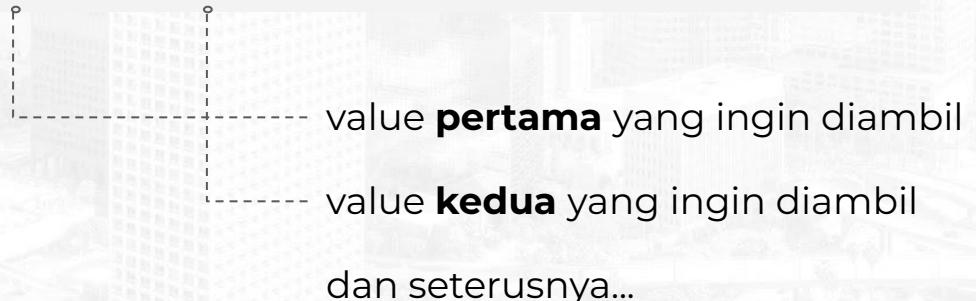
Single Filtering

SQL Analogi - IN

untuk **filter multiple value** dalam satu syntax

Format Syntax

```
df[ df['column_name'].isin( [ value_1, value_2, value_3, .... value_n ] ) ]
```



Single Filtering SQL Analogi - IN

untuk **filter multiple value** dalam satu syntax

Contoh 1 - soal

Lakukan filtering untuk mengambil data yang memiliki nama-nama berikut

- Reynard Ramadhansyah
- Putu Adinda
- Firman Reza
- Puspa Fariza
- Nadia Persadani

Single Filtering SQL Analogi - IN

untuk **filter multiple value** dalam satu syntax

Contoh 1 - jawaban

```
df_customers[df_customers['name'].isin(['Reynard Ramadhansyah', 'Putu Adinda',
                                         'Firman Reza','Puspa Fariza','Nadia Persadani'])]
```

	registration_id	name		email	phone	age	active
15	RK20202358	Reynard Ramadhansyah	reynardramadhansyah@gmail.com	6.289624e+09	34.0	True	
61	RK20202401	Nadia Persadani	nadiapersadani@roketmail.com	6.287834e+09	25.0	True	
68	RK20202408	Puspa Fariza	puspafariza@yahoo.com	6.285335e+09	34.0	True	
94	RK20202432	Firman Reza	firmanreza@gmail.com	6.286084e+09	28.0	True	
99	RK20202437	Putu Adinda	putuadinda@yahoo.com	6.288323e+09	38.0	True	

Single Filtering SQL Analogi - IN

untuk **filter multiple value** dalam satu syntax

Contoh 2 - soal

Lakukan filtering untuk mengambil data customer yang berasal dari

- Sukabumi
- Cirebon
- Tasikmalaya
- Singkawang

Single Filtering SQL Analogi - IN

untuk **filter multiple value** dalam satu syntax

Contoh 2 - jawaban

```
df_customers[df_customers['city'].isin(['Sukabumi','Cirebon','Tasikmalaya','Singkawang'])]
```

	registration_id	name	email	phone	age	active	city	province
18	RK20202361	Wahyu Hayati	wahyuhayati@gmail.com	6.289627e+09	23.0	True	Cirebon	Jawa Barat
30	RK20202373	Herdaru Junaidi	herdarujunaidi@outlook.com	6.287834e+09	50.0	True	Sukabumi	Jawa Barat
50	RK20202391	Anugrah Sastra	anugrahsastra@outlook.com	6.287899e+09	35.0	True	Cirebon	Jawa Barat
58	RK20202398	Anastasia Primanelza	anastasiaprimanelza@roketmail.com	6.287853e+09	40.0	True	Tasikmalaya	Jawa Barat
64	RK20202404	Amadea Octaza	amadeaoctaza@roketmail.com	6.287814e+09	17.0	True	Cirebon	Jawa Barat
91	RK20202431	Prima Hermawan	primahermawan@rocketmail.com	6.281856e+09	50.0	True	Singkawang	Kalimantan Barat
99	RK20202437	Putu Adinda	putuadinda@yahoo.com	6.288323e+09	38.0	True	Tasikmalaya	Jawa Barat

YUK LATIHAN !!!

Filter Kasus: Kedua

Carilah customer yang **TIDAK memiliki data nomor telefon**

atau

customer yang **data umur TIDAK diketahui**

Single Filtering SQL Analogi - IS NULL

untuk **filter berdasarkan NULL value** dari suatu kolom

Format Syntax

```
df[ df['column_name'].isnull() ]
```

Single Filtering SQL Analogi - IS NULL

untuk **filter berdasarkan NULL value** dari suatu kolom

Contoh Soal 1

Carilah customer yang **tidak memiliki data nomor telepon**

```
df_customers[df_customers['phone'].isnull()]
```

	registration_id	name	email	phone	age	active	city	province
14	RK20202357	Lintang Bertauli	lintangbertauli@gmail.com	NaN	33.0	True	Tanjungpinang	Kepulauan Riau
24	RK20202367	Ramanta Pasha	ramantapasha@gmail.com	NaN	49.0	True	Sungai Penuh	Jambi
31	RK20202374	Eka Tiodimar	ekatio@yahoo.com	NaN	NaN	True	Kota Administrasi Jakarta Utara	Jakarta

Single Filtering SQL Analogi - IS NULL

untuk **filter berdasarkan NULL value** dari suatu kolom

Contoh Soal 2

Carilah customer yang **tidak memiliki data umur**

```
df_customers[df_customers['age'].isnull()]
```

registration_id	name	email	phone	age	active	city	province
31	RK20202374	Eka Tiodimar	ekatio@yahoo.com	NaN	NaN	True	Kota Administrasi Jakarta Utara

Single Filtering

SQL Analogi - IS NOT NULL

untuk **filter berdasarkan NOT NULL value** dari suatu kolom

Format Syntax

```
df[ df['column_name'].notnull() ]
```

Single Filtering

SQL Analog - IS NOT NULL

untuk **filter berdasarkan NOT NULL value** dari suatu kolom

Contoh Soal 1

Carilah customer yang **memiliki data umur** saja

```
df_customers[df_customers['phone'].notnull()]
```

	registration_id	name	email	phone	age	active	city	province
0	RK20202344	Fakhrul Fandy Widyaningtias	fakhrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False	Tanjungpinang	Kepulauan Riau
1	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True	Bogor	Jawa Barat
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True	Kota Administrasi Jakarta Pusat	Jakarta
3	RK20202347	Singgih Kharisma	singgikhharisma@hotmail.com	6.285245e+09	49.0	True	Metro	Lampung
4	RK20202348	Jeremiah Dinanti	jeremia@rakamin.com	6.285245e+09	25.0	False	Malang	Jawa Timur

YUK LATIHAN !!!

Multi-Column Filter

Digunakan untuk melakukan **filtering dengan berbagai kondisi**, sesuai kebutuhan insight yang ingin didapatkan

Contoh Kasus

Marketing manager ingin membuat campaign untuk produk buku sekolah, dia membutuhkan data customer yang memenuhi kondisi berikut:

- Berasal dari Jawa Barat
- Umur dibawah 20 tahun
- Memiliki data nomor telepon, untuk selanjutnya dihubungi via sms

Multi-Column Filter

Format Syntax

untuk logic “**AND**” semua

```
df[ (kondisi_pertama) & (kondisi kedua) & .... (kondisi_ke_N)  
]
```

untuk logic “**OR**” semua

```
df[ (kondisi_pertama) | (kondisi kedua) | .... (kondisi_ke_N) ]
```

saat menggunakan logic “**OR**” dan “**AND**” bersamaan,
disarankan menggunakan double tanda kurung pada kondisi OR seperti berikut

```
df[ ( (kondisi_pertama) | (kondisi kedua) ) & (kondisi_ke_3) ... & (kondisi_ke_N)  
]
```

Multi-Column Filter

Contoh - Soal

Marketing manager ingin membuat campaign untuk produk buku sekolah, dia membutuhkan data customer yang memenuhi kondisi berikut:

- Berasal dari Jawa Barat
- Umur dibawah 20 tahun
- Memiliki data nomor telepon, untuk selanjutnya dihubungi via sms

Multi-Column Filter

Contoh - Jawaban

```
df_customers[ (df_customers['province'] == 'Jawa Barat' )
    & (df_customers['age'] < 20)
    & (df_customers['phone'].notnull() )]
```

	registration_id	name		email	phone	age	active	city	province
23	RK20202366	Muhamad Pramanda		muhamadpramanda@outlook.com	6.289633e+09	16.0	True	Depok	Jawa Barat
28	RK20202371	Fahmi Haq		fahmi@yahoo.com	6.287833e+09	17.0	True	Depok	Jawa Barat
34	RK20202377	Ismail Wardhani		ismailwardhani@hotmail.com	6.287835e+09	18.0	True	Bandung	Jawa Barat
38	RK20202381	Jova Imania		jovaimania@gmail.com	6.284688e+08	17.0	True	Depok	Jawa Barat
64	RK20202404	Amadea Octaza		amadeaoctaza@rocketmail.com	6.287814e+09	17.0	True	Cirebon	Jawa Barat
70	RK20202410	Hilary Adha		hilaryadha@yahoo.com	6.285321e+09	15.0	True	Bogor	Jawa Barat
74	RK20202414	Moerdiati Ramsihs		moerdiatiramsihs@rocketmail.com	6.285295e+09	16.0	True	Banjar	Jawa Barat
87	RK20202427	Janiar Satria		janiarsatria@outlook.com	6.285210e+09	17.0	True	Cimahi	Jawa Barat

YUK LATIHAN !!!

Filtering Lanjut

DataFrame

SQL Analogi LIKE

Filtering Sebelumnya

SQL Syntax	Pandas Code	Kegunaan
<column> = a	df[df['column'] == a] or != / > / >= / < / <=	Filter berdasarkan eksak nilai
<column> in (a, b, c)	df[df['column'].isin([a, b, c])]	Filter multiple eksak nilai
<column> is not null	df[df['column'].notnull()]	Filter supaya tidak ada null
<column> is null	df[df['column'].isnull()]	Filter supaya null saja

Use Case

Gimana cara filter untuk mengambil data yang memiliki **email ber-domain rakamin.com ?**

atau

filter customer yang **memiliki nama agus?**

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Format Syntax

```
df[ df['column_name'].str.contains('pattern') ]
```

syarat : kolom harus bertipe string

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh pattern - 1

email	pattern	apakah pattern match?
salihah@outlook.com		False
sutrisno@outlook.com		False
puspafariza@yahoo.com	yahoo.com	True
haikal@roketmail.com		False

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh pattern - 2

phone	pattern	apakah pattern match?
6285244922		False
6287840324	provider XL (62878xxxxx)	True
6285249258		False
6287892605		True

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh Soal 1

Lakukan filtering untuk mengambil data customer yang **mengandung nama agus**

```
df_customers[df_customers['name'].str.contains('agus')]
```

	registration_id	name	email	phone	age	active	city	province
96	RK20202434	Bagus Maulana	bagusmaulana@outlook.com	6.288953e+09	54.0	True	Batam	Kepulauan Riau

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh Soal 2

Lakukan filtering untuk mengambil customer yang memiliki **email berdomain yahoo.com**

```
df_customers[df_customers['email'].str.contains('yahoo.com')]
```

	registration_id	name	email	phone	age	active	city	province
1	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True	Bogor	Jawa Barat
10	RK20202353	Bob Arisa	bobarisa@yahoo.com	6.281909e+09	40.0	True	Salatiga	Jawa Tengah
17	RK20202360	Rifqy Butar-butar	rifqybutar-butar@yahoo.com	6.289626e+09	30.0	True	Semarang	Jawa Tengah
19	RK20202362	Andi Mayasopha	andimayasopha@yahoo.com	6.289628e+09	36.0	True	Magelang	Jawa Tengah
27	RK20202370	Doni Teddo Prasetyani	doniteddoprasetyani@yahoo.com	6.287833e+09	30.0	True	Batam	Kepulauan Riau
28	RK20202371	Fahmi Haq	fahmi@yahoo.com	6.287833e+09	17.0	True	Depok	Jawa Barat

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh Soal 3

Lakukan filtering untuk mengambil customer yang **phone mengandung digit 860**

```
df_customers[df_customers['phone'].str.contains('860')]
```

```
AttributeError                                 Traceback (most recent call last)
<ipython-input-10-d81fa07234c9> in <module>
----> 1 df_customers[df_customers['phone'].str.contains('860')]
~\AppData\Roaming\Python\Python38\site-packages\pandas\core\generic.py in __
 5485      ):
 5486          return self[name]
-> 5487          return object.__getattribute__(self, name)
 5488
 5489      def __delitem__(self, name):
 5490          if name in self._info_axis:
```

ERROR:
STR CONTAINS HANYA BERLAKU
UNTUK TIPE DATA STRING

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh Soal 3

ERROR SOLUSI : Ubah Tipe data dahulu ke string

Lakukan filtering untuk mengambil customer yang **phone mengandung digit 860**

```
df_customers[df_customers['phone'].astype('str').str.contains('860')]
```

	registration_id	name	email	phone	age	active	city	province
57	RK20202397	Anindyanti Bayhacki	anindyantibayhacki@yahoo.com	6.287860e+09	21.0	True	Cimahi	Jawa Barat
94	RK20202432	Firman Reza	firmanreza@gmail.com	6.286084e+09	28.0	True	Batam	Kepulauan Riau

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh pattern - 3

email	pattern	apakah pattern match?
salihah@outlook.com		True
sutrisno@outlook.com		True
puspafariza@yahoo.com	DIMULAI dengan huruf s	False
haikal@roketmail.com		False

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Format Syntax

Pattern **DIMULAI** dengan: `df[df['column_name'].str.contains('^pattern')]`

Pattern **DIAKHIRI** dengan: `df[df['column_name'].str.contains('pattern$')]`

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh Soal 1

Lakukan filtering untuk mengambil data customer memiliki **phone dengan provider XL**

```
df_customers[df_customers['phone'].astype('str').str.contains('^62878')]
```

registration_id	name	email	phone	age	active	city	province
RK20202405	Auzy Narendra	auzynarendra@outlook.com	6.287807e+09	35.0	True	Balikpapan	Kalimantan Timur
RK20202401	Nadia Persadani	nadiapersadani@roketmail.com	6.287834e+09	25.0	True	Surabaya	Jawa Timur
RK20202386	Yusuf Tilasnuari	yusuftilasnuari@hotmail.com	6.287888e+09	31.0	True	Malang	Jawa Timur
RK20202395	Indah Khairunisa	indahkhairunisa@roketmail.com	6.287873e+09	40.0	True	Salatiga	Jawa Tengah

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh Soal 2

Lakukan filtering untuk mengambil customer yang memiliki **nama berakhiran “ana”**

```
df_customers[df_customers['name'].str.contains('ana$')]
```

registration_id	name	email	phone	age	active	city	province
RK20202434	Bagus Maulana	bagusmaulana@outlook.com	6.288953e+09	54.0	True	Batam	Kepulauan Riau
RK20202440	Amalia Azizah Putriana	amaliaazizahputriana@yahoo.com	6.283542e+09	57.0	True	Kota Administrasi Jakarta Barat	Jakarta

Single Filtering

SQL Analogi - LIKE %

untuk **filter berdasarkan pattern/pola** dari suatu kolom

Contoh Soal 3

Lakukan filtering untuk mengambil customer yang memiliki **nama berawalan “andi”**

```
df_customers[df_customers['name'].str.lower().str.contains('^andi')]
```

registration_id	name	email	phone	age	active	city	province
RK20202362	Andi Mayasopha	andimayasopha@yahoo.com	6.289628e+09	36.0	True	Magelang	Jawa Tengah

YUK LATIHAN !!!

Menghapus Row **DataFrame**

Hapus Baris

Filter baris

(**mengambil baris** dengan nilai tertentu saja)

Syntax

```
df[ df['column'] == a ]
```

```
df[ df['column'].isin( [a, b, c] ) ]
```

```
df[ df['column'].notnull() ]
```

```
df[ df['column'].isnull() ]
```

```
df[ df['column'].str.contains('xxxx') ]
```

Hapus baris

(**Menghapus baris** dengan nilai tertentu)

Syntax

```
df[ ~(df['column'] == a) ]
```

```
df[ ~(df['column'].isin( [a, b, c] )) ]
```

```
df[ ~(df['column'].notnull()) ]
```

```
df[ ~(df['column'].isnull()) ]
```

```
df[ ~(df['column'].str.contains('xxxx')) ]
```

Hapus Baris

Contoh 1

Hapus baris pada **kolom province - Jawa Barat**

```
df_customers[~(df_customers['province']=='Jawa Barat')]
```

	registration_id	name	email	phone	age	income_group	active	city	province
0	RK20202344	Fakhru Fandy Widyaningtias	fakhruflandywidyaningtias@outlook.com	6.285244e+09	27.0		2	False	Tanjungpinang Kepulauan Riau
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0		1	True	Kota Administrasi Jakarta Pusat Jakarta
3	RK20202347	Singgih Kharisma	singgikhkharisma@hotmail.com	6.285245e+09	49.0		4	True	Metro Lampung
4	RK20202348	Jeremiah Dinanti	jeremia@rakamin.com	6.285245e+09	25.0		2	False	Malang Jawa Timur

Hapus Baris

Contoh 2

Hapus baris dimana **email tidak berdomain outlook**

```
df_customers[~(df_customers['email'].str.contains('outlook'))]
```

	registration_id	name	email	phone	age	income_group	active	city	province
1	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0		3	True	Bogor Jawa Barat
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0		1	True	Kota Administrasi Jakarta Pusat Jakarta
3	RK20202347	Singgih Kharisma	singgihkharisma@hotmail.com	6.285245e+09	49.0		4	True	Metro Lampung
4	RK20202348	Jeremiah Dinanti	jeremia@rakamin.com	6.285245e+09	25.0		2	False	Malang Jawa Timur
5	RK20202349	Ridhwan Noerani	ridhwannoerani@hotmail.com	6.285246e+09	43.0		4	False	Kota Administrasi Jakarta Pusat Jakarta

YUK LATIHAN !!!

Pandas DataFrame & Transformation

- | | |
|--|-------------------------|
|   | Pengenalan DataFrame |
|   | Pengenalan Pandas |
|   | Membuat DataFrame |
|   | Operasi Dasar DataFrame |
|   | Mengubah Tipe Data |
|   | Filtering Column |
|   | Membuat Column Baru |
|   | Menghapus Column |
|   | Menghapus Row |
|   | Sorting Row |
|   | Filtering Row |
|   | Export DataFrame |

Export DataFrame **ke CSV atau Excel**

Export Data Frame to Excel

Format Syntax

```
df.to_excel('nama_file.xlsx', index = False)
```

Isi `nama_file` sesuai
kebutuhan/context

Digunakan supaya
index pada data frame
tidak masuk ke file Excel

Export Data Frame to Excel

Contoh

Jika **tanpa index=False**

```
df_analisis.to_excel('data_hasil_analisis_tipe_1.xlsx')
```

	A	B	C	D	E	F
1		registration_id	name	email	phone	age
2	0	RK20202376	Rinaldy Octaza	rinaldyoctaza	6287834541	35
3	1	RK20202362	Andi Mayasophia	andimayasop	6289628397	36
4	2	RK20202385	Geraldi Lalo	geraldilalo@h	6287885801	50
5	3	RK20202366	Muhamad Pram	muhammadpra	6289633249	16
6	4	RK20202357	Lintang Bertauli	lintangbertauli@gmail.com		33
7	5	RK20202351	Dikposa Yosua K	dikposayosua	6281907247	28
8	6	RK20202356	Hilman Azalia	hilmanazalia@	6281911692	17
9	7	RK20202381	Jova Imania	jovaimania@	628468802	17

INDEX dari DataFrame masuk ke Excel Data

Export Data Frame to Excel

Contoh

Jika **Menggunakan index=False**

```
df_analisis.to_excel('data_hasil_analisis_tipe_2.xlsx',index=False)
```

	A	B	C	D	E
1	registration_id	name	email	phone	age
2	RK20202376	Rinaldy Octaza	rinaldyoctaza@	6,29E+09	35
3	RK20202362	Andi Mayasoph	andimayasoph	6,29E+09	36
4	RK20202385	Geraldi Lalo	geraldilalo@hc	6,29E+09	50
5	RK20202366	Muhamad Prar	muhamadpran	6,29E+09	16
6	RK20202357	Lintang Bertaui	lintangbertauli@gmail.co		33
7	RK20202351	Dikposa Yosua	dikposayosuak	6,28E+09	28
8	RK20202356	Hilman Azalia	hilmanazalia@	6,28E+09	17
9	RK20202381	Jova Imania	jovaimania@g	6,28E+08	17
10	RK20202365	Thomas Geryar	thomasgeryar	6,29E+09	40
11	RK20202383	Garin Sugiyanto	garinsugiyanto	6,29E+09	33

Export Data Frame to CSV

Format Syntax

```
df.to_csv('nama_file.csv', index=False)
```

YUK LATIHAN !!!

Pandas DataFrame & Transformation

- | | |
|---|--|
|   Pengenalan DataFrame |   Membuat Column Baru |
|   Pengenalan Pandas |   Menghapus Column |
|   Membuat DataFrame |   Menghapus Row |
|   Operasi Dasar DataFrame |   Sorting Row |
|   Mengubah Tipe Data |   Filtering Row |
|   Filtering Column |   Export DataFrame |

Terima Kasih!



M Ramadiansyah

Machine Learning Engineer



Muhammad Ramadiansyah

<https://id.linkedin.com/in/muhammad-ramadiansyah>