



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

DataFrame Aggregation



Topik *Data Processing*

Topik 1 :

Pandas Dataframe and Transformation

DONE

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

NOW

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

DataFrame Aggregation

- ☐ Pengenalan Aggregation
- ☐ GroupBy pada Pandas
- ☐ Reset Index DataFrame
- ☐ Mengubah Nama Column
- ☐ Pivot Table Aggregation
- ☐ Pivot Table Reshape

Hands-On Requirement:

Hands - On :

Data Processing - DataFrame Aggregation.ipynb

Dataset : [data_rakamin_customer.xlsx](#)

**Klik di sini
untuk akses dataset dan handsOn
Jupyter Notebook**

Dataset Overview

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

Informasi dasar dari customer Rakamin

Dataset Overview

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

Website Rakamin Customer Behaviour:

- Total Visit
- Recency (Kapan terakhir open web)
- Purchase Rate : $\text{Total Purchase} / \text{Total Visit}$
- Purchase Amount: Jumlah pembelian
- CS Call Count: Total menelpon ke CS

Dataset Overview

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

Campaign Activities: apakah Customer menerima ajakan dari campaign yang dibuat.

Total sudah ada 5 campaign berjalan yang dikirim ke setiap customer

DataFrame Aggregation



Pengenalan Aggregation



Mengubah Nama Column



GroupBy pada Pandas



Pivot Table Aggregation



Reset Index DataFrame



Pivot Table Reshape

Pengenalan **Data Aggregation**

Apa itu *Aggregation*?

Mengelompokan data berdasarkan Kolom tertentu dan mencari nilai statistiknya

nama	umur	nomor sepatu	hobi
Dadan	60	40	Memasak
Firman	15	39	Memasak
Ujang	33	43	Renang
Heru	17	41	Renang
Andaru	26	41	Renang
Yonathan	55	40	Musik
Ilham	44	44	Musik
Joshua	22	39	Musik
Petra	37	40	Musik

Pertanyaan

- Berapa **rata-rata umur** tiap grup hobi?
 - memasak = $(60+15)/2$
 - Renang = $(33+17+26)/3$
 - Musik = $(55+44+22+37)/4$

Group berdasarkan Hobi

DataFrame Aggregation



Pengenalan Aggregation



GroupBy pada Pandas



Reset Index DataFrame



Mengubah Nama Column



Pivot Table Aggregation



Pivot Table Reshape

GroupBy dan Aggregation **Pandas**

Hitung `total registration_id` per **Province**

SQL Query

```
SELECT province, count(registration_id) statistik
FROM data_rakamin_customers
GROUP BY province
      by parameter
```

Python Code

```
data_rakamin_customers.groupby('province').agg( {'registration_id': 'count'}
)
                        by parameter                        statistik
```

GroupBy - Satu By Parameter

Satu Kolom dan Satu Statistik

Format Syntax

```
df.groupby('column_name').agg({'column': 'statistics'})
```

nama
dataframe

nama	umur	nomor sepatu	hobi
Dadan	60	40	Memasak
Firman	15	39	Memasak
Ujang	33	43	Renang
Heru	17	41	Renang
Yonathan	55	40	Musik
Ilham	44	44	Musik

GroupBy - Satu By Parameter

Satu Kolom dan Satu Statistik

Format Syntax

```
df.groupby('column_name').agg({'column': 'statistics'})
```

by parameter
kolom yang
menjadi group



nama	umur	nomor sepatu	hobi
Dadan	60	40	Memasak
Firman	15	39	Memasak
Ujang	33	43	Renang
Heru	17	41	Renang
Yonathan	55	40	Musik
Ilham	44	44	Musik

GroupBy - Satu By Parameter

Satu Kolom dan Satu Statistik

Format Syntax

```
df.groupby('column_name').agg({'column': 'statistics'})
```

Kolom yang akan kita cari statistik aggregate nya

nama	umur	nomor sepatu	hobi
Dadan	60	40	Memasak
Firman	15	39	Memasak
Ujang	33	43	Renang
Heru	17	41	Renang
Yonathan	55	40	Musik
Ilham	44	44	Musik

GroupBy - Satu By Parameter

Satu Kolom dan Satu Statistik

Format Syntax

```
df.groupby('column_name').agg({'column': 'statistics'})
```

statistics	kegunaan
count	menghitung total baris
nunique	menghitung total baris yang unique
mean	rata-rata dari kolom
median	median data dari kolom
min	nilai terkecil dari kolom
max	nilai maksimal dari kolom

GroupBy - Satu By Parameter

Satu Kolom dan Satu Statistik

Contoh 1 Ada berapa customer (registration_id) per province?

```
df_customers.groupby('province').agg({'registration_id' : 'count'})
```

province	registration_id
Jakarta	13
Jambi	2
Jawa Barat	22
Jawa Tengah	14
Jawa Timur	17
Kalimantan Barat	6
Kalimantan Selatan	6

GroupBy - Satu By Parameter

Satu Kolom dan Satu Statistik

Contoh 2 Ada berapa **UNIQUE** customer (registration_id) per province?

```
df_customers.groupby('province').agg({'registration_id' : 'nunique'})
```

province	registration_id
Jakarta	12
Jambi	2
Jawa Barat	21
Jawa Tengah	13
Jawa Timur	16
Kalimantan Barat	4
Kalimantan Selatan	6

GroupBy - Satu By Parameter

Satu Kolom dan Satu Statistik

Contoh 3 Berapa **rata-rata umur** per province?

```
df_customers.groupby('province').agg({'age' : 'mean'})
```

province	age
Jakarta	33.166667
Jambi	44.500000
Jawa Barat	27.090909
Jawa Tengah	31.357143
Jawa Timur	37.470588
Kalimantan Barat	40.500000
Kalimantan Selatan	39.000000

GroupBy - Satu By Parameter

Satu Kolom dan Satu Statistik

Contoh 4 Berapa **rata-rata umur** per **income_group**?

```
df_customers.groupby('income_group').agg({'age' : 'mean'})
```

	age
class	
1	17.894737
2	25.807692
3	35.407407
4	43.851852
5	51.571429

GroupBy - Satu By Parameter

Satu Kolom dan Satu Statistik

Contoh 5 Berapa **median** website_visit_count per income_group?

```
df_customers.groupby('income_group').agg({'website_visit_count' : 'median'})
```

website_visit_count	
class	
1	47.0
2	54.5
3	36.0
4	58.0
5	46.0

GroupBy - Satu By Parameter

Satu Kolom dan Multiple Statistik

Format Syntax

```
df.groupby('column_name').agg({'column': ['statistik_1', 'statistik_2', .....]})
```

stats yang akan diambil
dimasukan ke dalam LIST

GroupBy - Satu By Parameter

Satu Kolom dan Multiple Statistik

Contoh 1 Berapa **rata-rata** dan **median** umur para customer per province?

```
df_customers.groupby('province').agg({'age' : ['mean', 'median']})
```

province	age	
	mean	median
Jakarta	33.166667	30.0
Jambi	44.500000	44.5
Jawa Barat	27.090909	22.0
Jawa Tengah	31.357143	33.5
Jawa Timur	37.470588	42.0
Kalimantan Barat	40.500000	50.0
Kalimantan Selatan	39.000000	40.0

GroupBy - Satu By Parameter

Satu Kolom dan Multiple Statistik

Contoh 2 Berapa **rata-rata** , **median** , **termuda** dan **tertua** umur per province?

```
df_customers.groupby('province').agg({'age' : ['mean', 'median', 'min', 'max']})
```

province	age			
	mean	median	min	max
Jakarta	33.166667	30.0	15.0	57.0
Jambi	44.500000	44.5	40.0	49.0
Jawa Barat	27.090909	22.0	15.0	50.0
Jawa Tengah	31.357143	33.5	15.0	42.0
Jawa Timur	37.470588	42.0	17.0	48.0
Kalimantan Barat	40.500000	50.0	15.0	54.0
Kalimantan Selatan	39.000000	40.0	24.0	48.0

GroupBy - Satu By Parameter

Satu Kolom dan Multiple Statistik

Contoh 3 Berapa **rata-rata** , dan **maksimum** age per website_cs_call_count?

```
df_customers.groupby('website_cs_call_count').agg({'age' : ['mean', 'max']})
```

website_cs_call_count	age	
	mean	max
0	33.240000	57.0
1	28.714286	47.0
2	35.125000	54.0
3	33.562500	50.0

YUK LATIHAN !!!

GroupBy - Satu By Parameter

Multiple Kolom dan Multiple Statistik

Format Syntax

```
df.groupby('column_name').agg( {'column_1': ['statistik_1', 'statistik_2', .....],  
                                'column_2': ['statistik_1', 'statistik_2', .....],  
                                :  
                                :  
                                :  
                                'column_N': ['statistik_1', 'statistik_2', .....] }  
)
```

GroupBy - Satu By Parameter

Multiple Kolom dan Multiple Statistik

Contoh 1

Hitunglah **banyaknya customer**, **rata-rata & median umur**, dan **website_purchase_rate minimum & maximum** per income_group?

```
df_customers.groupby('income_group').agg({'registration_id' : ['count'],
                                          'age' : ['mean', 'median'],
                                          'website_purchase_rate' : ['min', 'max']})
```

	registration_id	age		website_purchase_rate	
	count	mean	median	min	max
income_group					
1	19	17.894737	17.0	0.00	1.00
2	26	25.807692	25.5	0.00	0.90
3	27	35.407407	33.0	0.33	0.98
4	27	43.851852	44.0	0.00	1.00
5	8	51.571429	50.0	0.00	0.99

GroupBy - Satu By Parameter

Multiple Kolom dan Multiple Statistik

Contoh 2 Hitunglah **median website_recency**, rata-rata **website_purchase_amount**, dan rata-rata **website_visit_count** per city?

```
df_customers.groupby('city').agg({'website_recency' : ['median'],
                                  'website_purchase_amount' : ['mean'],
                                  'website_visit_count' : ['mean']})
```

	website_recency	website_purchase_amount	website_visit_count
	median	mean	mean
city			
Balikpapan	15.0	2.843333e+05	32.000000
Bandar Lampung	17.0	3.780000e+05	20.000000
Bandung	23.0	7.620000e+05	37.000000
Banjar	15.5	1.520000e+05	23.500000
Banjarbaru	8.5	1.601750e+06	37.750000

GroupBy - Multi By Parameter

Multiple Kolom dan Multiple Statistik

Format Syntax

```
df.groupby( [By Parameters] ).agg( {'column_1': [ 'statistik_1', 'statistik_2', ..... ],  
                                     'column_2': [ 'statistik_1', 'statistik_2', ..... ],  
                                     ⋮  
                                     'column_N': [ 'statistik_1', 'statistik_2', ..... ] }  
 )
```

GroupBy - Multi By Parameter

Multiple Kolom dan Multiple Statistik

Contoh 1

Hitunglah **banyaknya customer**, **rata-rata & median umur**, dan **website_purchase_rate minimum & maximum** per **province** dan **income_group**?

```
df_customers.groupby(['province', 'income_group']).agg({'registration_id' : ['nunique'],
                                                         'age' : ['mean', 'median'],
                                                         'website_purchase_rate' : ['min', 'max']})
```

province	income_group		registration_id	age	website_purchase_rate	
			nunique	mean	median	min
Jakarta	1	2	16.000000	16.0	0.69	0.78
	2	3	25.500000	27.0	0.00	0.90
	3	2	33.000000	33.0	0.60	0.77
	4	3	47.000000	49.0	0.00	0.41
	5	2	57.000000	57.0	0.00	0.99

GroupBy - Multi By Parameter

Multiple Kolom dan Multiple Statistik

Contoh 2

Hitunglah **median website_recency**, **rata-rata website_purchase_amount**, dan **rata-rata website_visit_count** per **province**, **city** dan **income_group**?

```
df_customers.groupby(['province', 'city', 'income_group']).agg({'website_recency' : ['median'],
                                                                'website_purchase_amount' : ['mean'],
                                                                'website_visit_count' : ['mean']})
```

			website_recency	website_purchase_amount	website_visit_count
			median	mean	mean
province	city	income_group			
Jakarta	Kota Administrasi Jakarta Barat	4	5.0	0.000000	70.000000
		5	19.0	99000.000000	44.000000
	Kota Administrasi Jakarta Pusat	1	19.0	276000.000000	30.000000
		2	57.0	0.000000	70.000000
		4	82.0	164000.000000	17.000000

YUK LATIHAN !!!

DataFrame Aggregation



Pengenalan Aggregation



GroupBy pada Pandas



Reset Index DataFrame



Mengubah Nama Column



Pivot Table Aggregation



Pivot Table Reshape

Reset Index **Data Frame**

Komponen DataFrame setelah dilakukan GroupBy

By Parameter menjadi Index
(Bukan Kolom)

Nama Kolom

By Parameter menjadi Index (Bukan Kolom)			Nama Kolom		
province	city	income_group	website_recency median	website_purchase_amount mean	website_visit_count mean
Jakarta	Kota Administrasi Jakarta Barat	4	5.0	0.000000	70.000000
		5	19.0	99000.000000	44.000000
	Kota Administrasi Jakarta Pusat	1	19.0	276000.000000	30.000000
		2	57.0	0.000000	70.000000
		4	82.0	164000.000000	17.000000
...
Kepulauan Riau	Tanjungpinang	2	75.0	612000.000000	51.000000
		3	20.0	274333.333333	51.666667

Reset Index DataFrame

Format Syntax

```
df = df.reset_index()
```

Reset Index DataFrame

Contoh 1 Resetlah index dari dataframe berikut

	No	Nama	Alamat	Usia
3	1	Andi Gunawan	Jakarta	21
8	2	Beni Tri	Bogor	24
21	3	Cinta Laudya	Bekasi	29
43	4	Deni Hermawan	Malang	24
99	5	Endang Gusti	Aceh	23

Reset Index DataFrame

Contoh 1 Resetlah index dari dataframe berikut

```
# reset index
data_reset = data.reset_index()

# cek hasil
data_reset
```

	index	No	Nama	Alamat	Usia
0	3	1	Andi Gunawan	Jakarta	21
1	8	2	Beni Tri	Bogor	24
2	21	3	Cinta Laudya	Bekasi	29
3	43	4	Deni Hermawan	Malang	24
4	99	5	Endang Gusti	Aceh	23

Reset Index DataFrame

Contoh 1 Resetlah index dari dataframe berikut

Before

	No	Nama	Alamat	Usia
3	1	Andi Gunawan	Jakarta	21
8	2	Beni Tri	Bogor	24
21	3	Cinta Laudya	Bekasi	29
43	4	Deni Hermawan	Malang	24
99	5	Endang Gusti	Aceh	23

After

	index	No	Nama	Alamat	Usia
0	3	1	Andi Gunawan	Jakarta	21
1	8	2	Beni Tri	Bogor	24
2	21	3	Cinta Laudya	Bekasi	29
3	43	4	Deni Hermawan	Malang	24
4	99	5	Endang Gusti	Aceh	23

Index akan menjadi kolom pada dataframe setelah dilakukan `reset_index()`

Reset Index DataFrame

Format Syntax 2

```
df = df.reset_index(drop=True)
```

Supaya setelah reset index tidak menjadi kolom di dataframe

Reset Index DataFrame

Contoh 1 Resetlah index dari dataframe berikut

```
# reset index
data_reset = data.reset_index()

# cek hasil
data_reset
```

	index	No	Nama	Alamat	Usia
0	3	1	Andi Gunawan	Jakarta	21
1	8	2	Beni Tri	Bogor	24
2	21	3	Cinta Laudya	Bekasi	29
3	43	4	Deni Hermawan	Malang	24
4	99	5	Endang Gusti	Aceh	23

VS

```
# reset index
data_reset2 = data.reset_index(drop=True)

# cek hasil
data_reset2
```

	No	Nama	Alamat	Usia
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

Reset Index DataFrame

Contoh 2 Resetlah index dari hasil groupby berikut

```
df_customers.groupby(['province', 'city', 'income_group']).agg({'website_recency' : ['median'],
                                                                'website_purchase_amount' : ['mean'],
                                                                'website_visit_count' : ['mean']})
```

			website_recency	website_purchase_amount	website_visit_count
			median	mean	mean
province	city	income_group			
Jakarta	Kota Administrasi Jakarta Barat	4	5.0	0.000000	70.000000
		5	19.0	99000.000000	44.000000
	Kota Administrasi Jakarta Pusat	1	19.0	276000.000000	30.000000
		2	57.0	0.000000	70.000000
		4	82.0	164000.000000	17.000000

```
# simpan hasil group by terlebih dahulu
```

[illegible]

Reset Index DataFrame

Contoh 2 Resetlah index dari hasil groupby berikut

```
# Lakukan reset index
df_grouped_reset = df_grouped.reset_index()
```

```
# cek dataframe setelah dilakukan reset index
df_grouped_reset
```

	province	city	income_group	website_recency	website_purchase_amount	website_visit_count
				median	mean	mean
0	Jakarta	Kota Administrasi Jakarta Barat	4	5.0	0.000000	70.000000
1	Jakarta	Kota Administrasi Jakarta Barat	5	19.0	99000.000000	44.000000
2	Jakarta	Kota Administrasi Jakarta Pusat	1	19.0	276000.000000	30.000000
3	Jakarta	Kota Administrasi Jakarta Pusat	2	57.0	0.000000	70.000000
4	Jakarta	Kota Administrasi Jakarta Pusat	4	82.0	164000.000000	17.000000

DataFrame Aggregation



Pengenalan Aggregation



GroupBy pada Pandas



Reset Index DataFrame



Mengubah Nama Column



Pivot Table Aggregation



Pivot Table Reshape

Mengubah **Nama Kolom**

satu kolom | beberapa kolom | semua kolom

Ubah Nama Satu Kolom

Format Syntax

```
df = df.rename(columns = { 'old_name' : 'new_name' })
```

Ubah Nama Satu Kolom

Contoh 1 Mengubah nama kolom `age` menjadi `usia_customer`

```
# Kita assign/simpan ke df_customers lagi supaya hasilnya permanent
```

```
df_customers = df_customers.rename(columns = {'age' : 'usia_customer'})
```

```
# Check apakah nama kolom sudah berubah
```

```
df_customers.head()
```

	registration_id	name	email	phone	usia_customer	active	city
0	RK20202344	Fakhrul Fandy Widyaningtias	fakhrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False	Tanjungpinang
1	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True	Bogor
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True	Kota Administrasi Jakarta Pusat
3	RK20202347	Singgih Kharisma	singgihkharisma@hotmail.com	6.285245e+09	49.0	True	Metro

Ubah Nama Satu Kolom

Contoh 2 Mengubah nama kolom `city` menjadi `kota_customer`

```
df_customers = df_customers.rename(columns = {'city' : 'kota_customer'})
```

```
df_customers.head() #cek 5 baris teratas untuk melihat hasilnya
```

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

Ubah Nama

Dua atau Lebih Kolom

Contoh

Mengubah nama kolom `registration_id` menjadi `ID`, dan `name` menjadi `nama_customers`

```
# Ubah dua kolom sekaligus
df_customers = df_customers.rename(columns = {'registration_id' : 'ID',
                                              'name' : 'nama_customers'})

df_customers.head()
```

	ID	nama_customers	email	phone	usia_customer	active	kota_customer
0	RK20202344	Fakhrul Fandy Widyaningtias	fakhrulfandywidyaningtias@outlook.com	6.285244e+09	27.0	False	Tanjungpinang
1	RK20202345	Ferdiansyah Amalina	ferdiansyahamalina@yahoo.com	6.285245e+09	31.0	True	Bogor
2	RK20202346	Ficky Rizkyananta	fickyrizkyananta@hotmail.com	6.285245e+09	15.0	True	Kota Administrasi Jakarta Pusat

Ubah Nama Semua Kolom

Format Syntax `df.columns = ['name_1', 'name_2', ... 'name_N']`

****** Syntax ini membuat semua nama kolom otomatis berubah permanent

Ubah Nama Semua Kolom

soal

(SOAL SEBELUMNYA)

Hitunglah **banyaknya customer**, **rata-rata & median umur**, dan **website_purchase_rate minimum & maximum** per **income_group**?

Kemudian ubah semua nama kolom dari hasil di atas!

```
df_grouped = df_customers.groupby('income_group').agg({'registration_id' : ['nunique'],
                                                         'age' : ['mean', 'median'],
                                                         'website_purchase_rate' : ['min', 'max']})
```

Hasil groupby disimpan dulu
ke dataframe baru

Ubah Nama Semua Kolom

Jawaban reset index supaya by parameter menjadi nama kolom

```
df_grouped = df_grouped.reset_index()
```

```
df_grouped
```

	income_group	registration_id	age		website_purchase_rate	
		nunique	mean	median	min	max
0	1	18	17.894737	17.0	0.00	1.00
1	2	25	25.807692	25.5	0.00	0.90
2	3	26	35.407407	33.0	0.33	0.98
3	4	26	43.851852	44.0	0.00	1.00
4	5	6	51.571429	50.0	0.00	0.99

Ubah Nama Semua Kolom

Jawaban mulai mengubah nama kolom hasil groupby

```
df_grouped = df_grouped.reset_index()
df_grouped.columns = ['income_group', 'total_customers', 'mean_umur', 'median_umur',
                      'purchase_rate_min', 'purchase_rate', 'max']
df_grouped
```

	income_group	total_customers	mean_umur	median_umur	purchase_rate_min	purchase_rate	max
0	0	1	18	17.894737	17.0	0.00	1.00
1	1	2	25	25.807692	25.5	0.00	0.90
2	2	3	26	35.407407	33.0	0.33	0.98
3	3	4	26	43.851852	44.0	0.00	1.00
4	4	5	6	51.571429	50.0	0.00	0.99

YUK LATIHAN !!!

DataFrame Aggregation



Pengenalan Aggregation



GroupBy pada Pandas



Reset Index DataFrame



Mengubah Nama Column



Pivot Table Aggregation



Pivot Table Reshape

Pivot Table

Data Frame

Data Aggregation (appendix) | Reshape Column & Row

Pivot Table

untuk Data Aggregation

Format Syntax

```
pd.pivot_table(df,  
               index = [by_parameters],  
               values = [column_1, column_2, ... column_N],  
               aggfunc = { 'column_1' : statistics,  
                           'column_2' : statistics,  
                           :  
                           :  
                           'column_N' : statistics  
                           }  
               )
```

Pivot Table

untuk Data Aggregation

Contoh 1

Hitunglah **banyaknya customer**, **rata-rata & median umur**, dan **website_purchase_rate minimum & maximum** per **income_group**?

```
pd.pivot_table(df_customers,
                index=['income_group'],
                values=['registration_id', 'age', 'website_purchase_rate'],
                aggfunc={'registration_id' : ['nunique'],
                        'age' : ['mean', 'median'],
                        'website_purchase_rate' : ['min', 'max']}
                )
```

Pivot Table

untuk Data Aggregation

Contoh 2

Hitunglah **banyaknya customer**, **rata-rata & median umur**, dan **website_purchase_rate minimum & maximum** per **province** dan **income_group**?

```
pd.pivot_table(df_customers,
                index=['province', 'income_group'],
                values=['registration_id', 'age', 'website_purchase_rate'],
                aggfunc={'registration_id' : ['nunique'],
                        'age' : ['mean', 'median'],
                        'website_purchase_rate' : ['min', 'max']}
                )
```

Pivot Table

untuk Reshape Data Frame

Biasa digunakan untuk mempermudah ketika analisis data

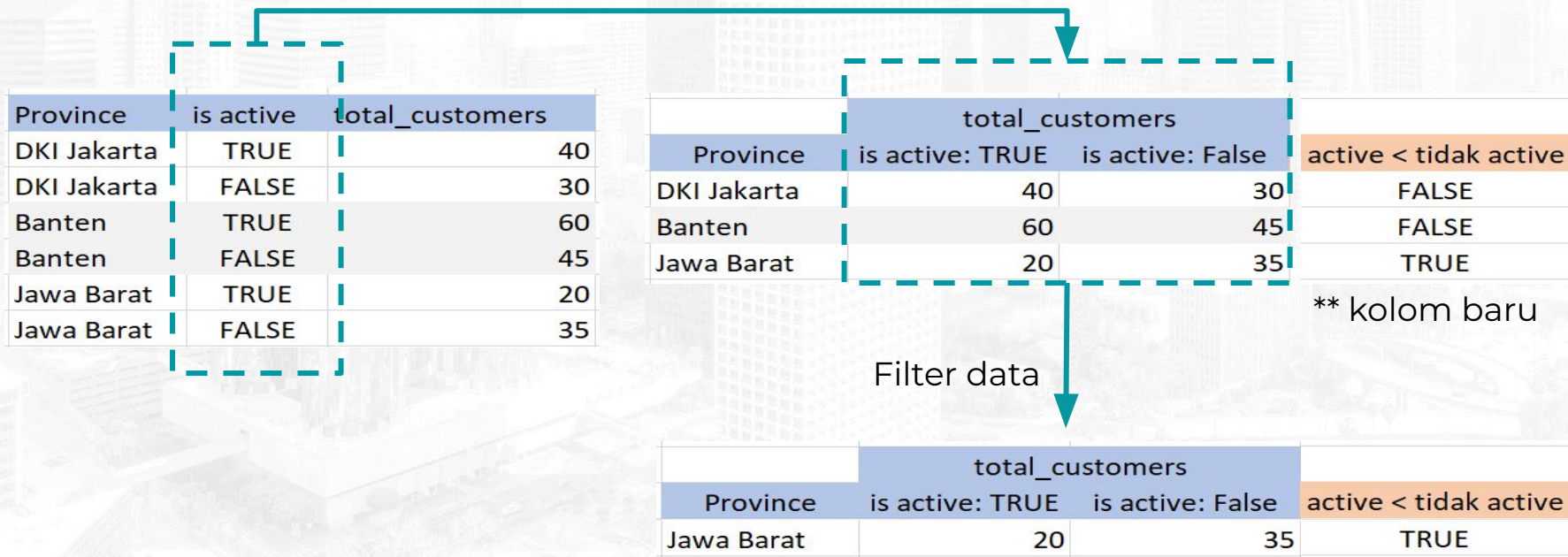
Contoh

Pada provinsi mana, total customer yang aktif lebih sedikit dibanding yang tidak aktif?

Province	is active	total_customers
DKI Jakarta	TRUE	40
DKI Jakarta	FALSE	30
Banten	TRUE	60
Banten	FALSE	45
Jawa Barat	TRUE	20
Jawa Barat	FALSE	35

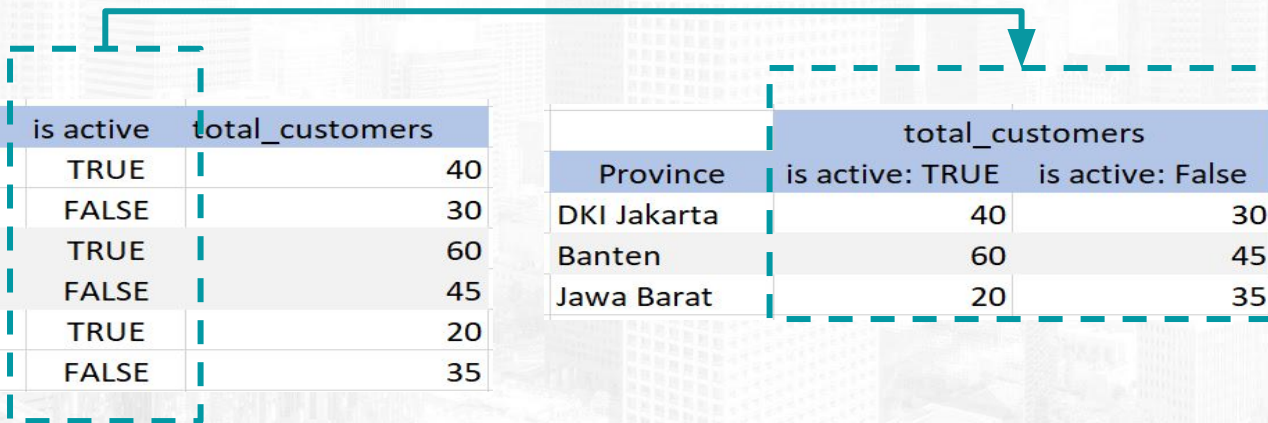
Pivot Table

untuk Reshape Data Frame



Pivot Table

untuk Reshape Data Frame



Province	is active	total_customers
DKI Jakarta	TRUE	40
DKI Jakarta	FALSE	30
Banten	TRUE	60
Banten	FALSE	45
Jawa Barat	TRUE	20
Jawa Barat	FALSE	35

	total_customers	
Province	is active: TRUE	is active: False
DKI Jakarta	40	30
Banten	60	45
Jawa Barat	20	35

```
pd.pivot_table(df,
                index = ['province'],
                columns = ['is_active'],
                values = ['total_customer'])
```

Pivot Table

untuk Reshape Data Frame

Contoh 1

```
df = pd.DataFrame({'province'      : ['DKI Jakarta', 'DKI Jakarta',  
                                     'Banten', 'Banten',  
                                     'Jawa Barat', 'Jawa Barat'],  
                  'is_active'      : [1, 0, 1, 0, 1, 0],  
                  'total_customers': [40, 30, 60, 45, 20, 35]})
```

df

	province	is_active	total_customers
0	DKI Jakarta	1	40
1	DKI Jakarta	0	30
2	Banten	1	60
3	Banten	0	45
4	Jawa Barat	1	20
5	Jawa Barat	0	35

Pivot Table

untuk Reshape Data Frame

Contoh 1

```
df_pivot = pd.pivot_table(df,
                           index = ['province'],
                           columns = ['is_active'],
                           values = ['total_customers'])

df_pivot
```

	total_customers	
	0	1
province		
Banten	45	60
DKI Jakarta	30	40
Jawa Barat	35	20

Pivot Table

untuk Reshape Data Frame

Contoh 2

STEP

1. Hitunglah jumlah customer yang accept campaign_1 dan tidak accept campaign_1 per **income_group**.

accept campaign_1: ketika kolom accept_campaign_1 bernilai True

tidak accept campaign_1: ketika kolom accept_campaign_1 bernilai False

2. Pada **income_group** berapa yang jumlah accept campaign lebih banyak dari yang tidak terima campaign?

Pivot Table

untuk Reshape Data Frame

```
df_grouped = df_customers.groupby(['income_group',
                                   'accept_campaign_1']).agg({'registration_id' : ['nunique']})
df_grouped = df_grouped.reset_index()
df_grouped.columns = ['income_group', 'accept_campaign_1', 'total_customers']
df_grouped
```

income_group	accept_campaign_1	total_custoemrs
1	False	7
1	True	11
2	False	13
2	True	12
3	False	12
3	True	14
4	False	15
4	True	11
5	False	2
5	True	4

income_group 1, 3 dan 5 memiliki jumlah yang accept campaign lebih banyak dibanding yang tidak accept campaign

Pivot Table

untuk Reshape Data Frame

```
pd.pivot_table(df_grouped,
                index = ['income_group'],
                columns = ['accept_campaign_1'],
                values = ['total_customers']
                )
```

Pivot Table untuk ubah accept_campaign_1 menjadi 2 kolom: False dan True

	total_customers	
	False	True
income_group		
1	7	11
2	13	12
3	12	14
4	15	11
5	2	4

Pivot Table

untuk Reshape Data Frame

```

df_pivot = pd.pivot_table(df_grouped,
                           index = ['income_group'],
                           columns = ['accept_campaign_1'],
                           values = ['total_customers']
                           )

df_pivot = df_pivot.reset_index()
df_pivot.columns = ['income_group', 'customer_non_accept', 'customer_accept']

df_pivot[df_pivot['customer_accept'] > df_pivot['customer_non_accept']]

```

	income_group	customer_non_accept	customer_accept
0	1	7	11
2	3	12	14
4	5	2	4

YUK LATIHAN !!!

Terima Kasih!



M Ramadiansyah

Machine Learning Engineer

finAcel



Muhammad Ramadiansyah

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