



# 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 Combination



# 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*

NOW

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

## Topik 2 : *DataFrame Aggregation*

DONE

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

# ***Hands-On Requirement:***

**Hands - On :**

**Data Processing - DataFrame Combination.ipynb**

**Dataset :**

1. **data\_rakamin\_customer.csv**

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



# DataFrame Combination

- ☐ Append Dictionary
- ☐ Append DataFrame
- ☐ Merge DataFrame

# DataFrame Combination



Append Dictionary



Append DataFrame



Merge DataFrame

# Append **Dictionary** ke **DataFrame**

# Append Dictionary

Menggabungkan Dictionary ke sebuah DataFrame

	no	hewan	Owner
0	1	ayam	dadang
1	2	bebek	didin
2	3	sapi	deden
3	4	kucing	dudin

df

```
{'no': 5, 'hewan': 'capung', 'Owner': 'adi'}
```

dictionary

**Format Code :**

```
df.append(dictionary, ignore_index=True)
```



# Append Dictionary

Menggabungkan Dictionary ke sebuah DataFrame

**Contoh Code :** Buat 1 dataframe dan 1 dictionary berikut secara manual, kemudian gabungkan menjadi satu dataframe

```
df = pd.DataFrame({'no' : [1 , 2 , 3 , 4] ,
                  'hewan' : ['ayam' , 'bebek' , 'sapi' , 'kucing'] ,
                  'Owner' : ['dadang' , 'didi' , 'deden' , 'dudin']
                  })
```

df

	no	hewan	Owner
0	1	ayam	dadang
1	2	bebek	didi
2	3	sapi	deden
3	4	kucing	dudin

# Append Dictionary

Menggabungkan Dictionary ke sebuah DataFrame

**Contoh Code :** Buat 1 dataframe dan 1 dictionary berikut secara manual, kemudian gabungkan menjadi satu dataframe

```
dictionary = {'no' : 5 ,  
             'hewan' : 'capung' ,  
             'Owner' : 'adi'  
            }  
  
dictionary  
  
{'no': 5, 'hewan': 'capung', 'Owner': 'adi'}
```

# Append Dictionary

Menggabungkan Dictionary ke sebuah DataFrame

**Contoh Code :** Buat 1 dataframe dan 1 dictionary berikut secara manual, kemudian gabungkan menjadi satu dataframe

```
df_result = df.append(dictionary, ignore_index=True)
df_result
```

	no	hewan	Owner
0	1	ayam	dadang
1	2	bebek	didin
2	3	sapi	deden
3	4	kucing	dudin
4	5	capung	adi

\*\* Hasil penggabungan muncul di baris paling akhir

# Append Dictionary

Menggabungkan Dictionary ke sebuah DataFrame

**Contoh Code 2 :** Kasus di mana Dictionary memiliki value berupa List

```
dictionary2 = {'no' : [5, 6] ,
               'hewan' : ['capung', 'gajah'] ,
               'Owner' : ['adi', 'ade', 'ado']
               }
```

dictionary2

```
{'no': [5, 6], 'hewan': ['capung', 'gajah'], 'Owner': ['adi', 'ade', 'ado']}
```

# Append Dictionary

Menggabungkan Dictionary ke sebuah DataFrame

**Contoh Code 2 :** Kasus di mana Dictionary memiliki value berupa List

```
df_result = df.append(dictionary2, ignore_index=True)
df_result
```

	no	hewan	Owner
0	1	ayam	dadan
1	2	bebek	didin
2	3	sapi	deden
3	4	kucing	dudin
4	[5, 6]	[capung, gajah]	[adi, ade, ado]

**\*\*** Walaupun Dictionary memiliki value List,  
Output append hanya menampilkan satu baris saja



# Append Dictionary

## Menggabungkan Dictionary ke sebuah DataFrame

**Contoh Code 3 :** Kasus di mana Dictionary memiliki value berupa kombinasi nilai

```
dictionary3 = {'no' : [5, 6, 8] ,
               'hewan' : ['capung', 'gajah', {'location' : 'ragunan',
                                                'name' : ['jerapah', 'singa']}] ,
               'Owner' : {'city' : 'bekasi',
                           'name' : 'adi',
                           'hobby' : ['memancing', 'menulis', 'menari'],
                           'cita-cita' : {'SD' : 'pilot',
                                           'SMP' : 'dokter',
                                           'SMA' : 'youtuber',
                                           'Sekarang' : 'Presiden'}}
               }

dictionary3
```

```
{'no': [5, 6, 8],
 'hewan': ['capung',
           'gajah',
           {'location': 'ragunan', 'name': ['jerapah', 'singa']}],
 'Owner': {'city': 'bekasi',
            'name': 'adi',
            'hobby': ['memancing', 'menulis', 'menari'],
            'cita-cita': {'SD': 'pilot',
                          'SMP': 'dokter',
                          'SMA': 'youtuber',
                          'Sekarang': 'Presiden'}}}
```

# Append Dictionary

Menggabungkan Dictionary ke sebuah DataFrame

**Contoh Code 3 :** Kasus di mana Dictionary memiliki value berupa kombinasi nilai

```
df_result = df.append(dictionary3, ignore_index=True)
df_result
```

	no	hewan	Owner
0	1	ayam	dadan
1	2	bebek	didin
2	3	sapi	deden
3	4	kucing	dudin
4	[5, 6, 8]	[capung, gajah, {'location': 'ragunan', 'name'...	{'city': 'bekasi', 'name': 'adi', 'hobby': ['m...

# Append Dictionary

## Aturan 1

- **Total keys** pada dictionary, **tidak harus sama dengan total kolom** di dataframe. dan akan menghasilkan NULL value

```
# Df berikut memiliki 3 kolom (no, hewan dan owner)

df = pd.DataFrame({'no'      : [1, 2, 3, 4],
                  'hewan'    : ['ayam', 'bebek', 'sapi', 'kucing'],
                  'Owner'    : ['dadang', 'didin', 'deden', 'dudin']
                  })

# Dictionary berikut hanya memiliki 2 keys (no dan hewan)

dictionary = {'no'      : 5,
              'hewan'   : 'capung'
              }
```

```
df.append(dictionary, ignore_index=True)
```

	no	hewan	Owner
0	1	ayam	dadang
1	2	bebek	didin
2	3	sapi	deden
3	4	kucing	dudin
4	5	capung	NaN

# Append Dictionary

## Aturan 2

- Jika **jumlah keys melebihi jumlah kolom** pada df, maka df akan memiliki tambahan kolom atas dictionary tersebut dan nilai baris lainnya menjadi NULL

```
# Df berikut memiliki 3 kolom
df = pd.DataFrame({'no' : [1, 2, 3, 4],
                  'hewan' : ['ayam', 'bebek', 'sapi', 'kucing'],
                  'owner' : ['dadan', 'didin', 'deden', 'dudin']
                  })

# Dictionary berikut memiliki 4 keys
dictionary = {'no' : 5,
              'hewan' : 'capung',
              'owner' : 'ade',
              'lokasi' : 'ragunan'
              }
```

```
df.append(dictionary, ignore_index=True)
```

	no	hewan	owner	lokasi
0	1	ayam	dadan	NaN
1	2	bebek	didin	NaN
2	3	sapi	deden	NaN
3	4	kucing	dudin	NaN
4	5	capung	ade	ragunan

# Append Dictionary

## Aturan 3

- Urutan keys tidak harus sama dengan urutan kolom pada df, yang terpenting adalah nama-nya bersesuaian.

```
# Df berikut memiliki kolom dengan urutan no, hewan dan owner
df = pd.DataFrame({'no' : [1, 2, 3, 4],
                  'hewan' : ['ayam', 'bebek', 'sapi', 'kucing'],
                  'owner' : ['dadan', 'didin', 'deden', 'dudin']
                  })

# Dictionary berikut memiliki 3 keys dengan urutan no, owner dan hewan
dictionary = {'no' : 5,
              'owner' : 'ade',
              'hewan' : 'capung'
              }
```

```
df.append(dictionary, ignore_index=True)
```

	no	hewan	owner
0	1	ayam	dadan
1	2	bebek	didin
2	3	sapi	deden
3	4	kucing	dudin
4	5	capung	ade



# **YUK LATIHAN !!!**

soal akan ditampilkan saat kelas :)

# DataFrame Combination



Append Dictionary



Append DataFrame



Merge DataFrame

# Append **DataFrame** ke **DataFrame**

# Append

Menggabungkan 2 atau lebih data frame

**Format Code :**

```
df.append(df_2)
```

# Append

## Menggabungkan 2 atau lebih data frame

**Contoh Code :** Buat 2 dataframe berikut secara manual, kemudian gabungkan menjadi satu dataframe

```
▼ df1 = pd.DataFrame({'no'      : [1 , 2 , 3 , 4 ] ,
                      'hewan'   : ['ayam' , 'bebek' , 'sapi' , 'kucing'] ,
                      'Owner'   : ['dadan' , 'didin' , 'deden' , 'dudin']
                      })

▼ df2 = pd.DataFrame({'no'      : [5 , 6 , 7] ,
                      'hewan'   : ['capung' , 'semut' , 'udang' ] ,
                      'Owner'   : ['adi' , 'ade' , 'ado']
                      })
```



# Append

## Menggabungkan 2 atau lebih data frame

**Contoh Code :** Buat 2 dataframe berikut secara manual, kemudian gabungkan menjadi satu dataframe

df1			
	no	hewan	Owner
0	1	ayam	dadan
1	2	bebek	didin
2	3	sapi	deden
3	4	kucing	dudin

df2			
	no	hewan	Owner
0	5	capung	adi
1	6	semut	ade
2	7	udang	ado



df_result = df1.append(df2) df_result			
	no	hewan	Owner
0	1	ayam	dadan
1	2	bebek	didin
2	3	sapi	deden
3	4	kucing	dudin
0	5	capung	adi
1	6	semut	ade
2	7	udang	ado

# **YUK LATIHAN !!!**

soal akan ditampilkan saat kelas :)

# DataFrame Combination



Append Dictionary



Append DataFrame



Merge DataFrame

# Merge **DataFrame**

# Merge Data Frame

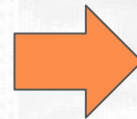
Proses menggabungkan 2 data frame menjadi 1 data frame berdasarkan kolom yang sama

df\_hobi

nama	hobi
Dadan	Memasak
Firman	Memasak
Ujang	Renang
Heru	Renang
Andaru	Renang

df\_umur

nama	umur
Dadan	60
Firman	15
Andaru	26
Yonathan	55
Ilham	44
Petra	37



nama	hobi	umur
Dadan	Memasak	60
Firman	Memasak	15
Ujang	Renang	NA
Heru	Renang	NA
Andaru	Renang	26

Left



# Merge Data Frame

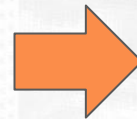
Proses menggabungkan 2 data frame menjadi 1 data frame berdasarkan kolom yang sama

df\_hobi

nama	hobi
Dadan	Memasak
Firman	Memasak
Ujang	Renang
Heru	Renang
Andaru	Renang

df\_umur

nama	umur
Dadan	60
Firman	15
Andaru	26
Yonathan	55
Ilham	44
Petra	37



nama	hobi	umur
Dadan	Memasak	60
Firman	Memasak	15
Andaru	Renang	26

**Inner**

# Merge Data Frame

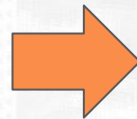
Proses menggabungkan 2 data frame menjadi 1 data frame berdasarkan kolom yang sama

df\_hobi

nama	hobi
Dadan	Memasak
Firman	Memasak
Ujang	Renang
Heru	Renang
Andaru	Renang

df\_umur

nama	umur
Dadan	60
Firman	15
Andaru	26
Yonathan	55
Ilham	44
Petra	37



nama	umur	hobi
Dadan	60	Memasak
Firman	15	Memasak
Andaru	26	Renang
Yonathan	55	NA
Ilham	44	NA
Petra	37	NA

Right

# Merge Data Frame

## Satu Kolom Matching

Format Syntax

### Pandas Merge

```
df_1.merge(df_2,
            left_on = col_1,
            right_on = col_2,
            how = 'inner'
            )
```

df\_1

col_1	columns
a	xx
b	xx
c	xx
d	xx
e	xx

df\_2

col_2	columns
a	yy
c	yy
d	yy
e	yy
f	yy
g	yy

# Merge Data Frame

## Satu Kolom Matching

### Contoh

```
df_users.merge(df_hobi,
               left_on = 'nama',
               right_on = 'nama',
               how = 'inner'
               )
```

df\_users

id	nama	umur
1	Ade	17
2	Hendra	25
3	Ammar	40
4	Andhika	10
5	Rezki	34
6	Fauzan	43
7	Choiri	34
8	Suprpto	22

df\_hobi

nama	hobi
Ade	memasak
Hendra	basket
Ammar	musik
Heri	gundu
Rezki	menari
Gunawan	berenang
Choiri	mendaki

# Merge Data Frame

## Satu Kolom Matching

### Contoh

```
df_users.merge(df_hobi,  
               left_on = 'nama',  
               right_on = 'nama',  
               how = 'inner'  
               )
```

df\_users

df\_hobi

id	nama	umur	hobi
1	Ade	17	memasak
2	Hendra	25	basket
3	Ammar	40	musik
5	Rezki	34	menari
7	Choiri	34	mendaki

# Merge Data Frame

## Satu Kolom Matching

### simplifikasi 1 :

Inner adalah default merge, maka bisa dihilangkan dalam syntax `how='inner'` untuk mempermudah

```
df_users.merge(df_hobi,
               left_on = 'nama',
               right_on = 'nama'
               )
```

### simplifikasi 2 :

Jika `left_on` dan `right_on` memiliki nama yang sama, maka cukup gunakan `on='column_name'`

```
df_users.merge(df_hobi,
               on = 'nama'
               )
```



# Merge Data Frame

## Satu Kolom Matching

Contoh

**LEFT JOIN**

```
df_users.merge(df_hobi,
               left_on = 'nama',
               right_on = 'nama',
               how = 'left'
               )
```

df\_users

id	nama	umur
1	Ade	17
2	Hendra	25
3	Ammar	40
4	Andhika	10
5	Rezki	34
6	Fauzan	43
7	Choiri	34
8	Suprpto	22

df\_hobi

nama	hobi
Ade	memasak
Hendra	basket
Ammar	musik
Heri	gundu
Rezki	menari
Gunawan	berenang
Choiri	mendaki

# Merge Data Frame

## Satu Kolom Matching

Contoh

**LEFT JOIN**

```
df_users.merge(df_hobi,
               left_on = 'nama',
               right_on = 'nama',
               how = 'left'
               )
```

df\_users

df\_hobi

id	nama	umur	hobi
1	Ade	17	memasak
2	Hendra	25	basket
3	Ammar	40	musik
4	Andhika	10	NaN
5	Rezki	34	menari
6	Fauzan	43	NaN
7	Choiri	34	mendaki
8	Suprapto	22	NaN

# Merge Data Frame

## Satu Kolom Matching

Contoh

**RIGHT JOIN**

```
df_users.merge(df_hobi,
               left_on = 'nama',
               right_on = 'nama',
               how = 'right'
               )
```

df\_users

id	nama	umur
1	Ade	17
2	Hendra	25
3	Ammar	40
4	Andhika	10
5	Rezki	34
6	Fauzan	43
7	Choiri	34
8	Suprpto	22

df\_hobi

nama	hobi
Ade	memasak
Hendra	basket
Ammar	musik
Heri	gundu
Rezki	menari
Gunawan	berenang
Choiri	mendaki



# Merge Data Frame

## Satu Kolom Matching

Contoh

**RIGHT JOIN**

```
df_users.merge(df_hobi,
               left_on = 'nama',
               right_on = 'nama',
               how = 'right'
               )
```

df\_users

df\_hobi

id	nama	umur	hobi
1.0	Ade	17.0	memasak
2.0	Hendra	25.0	basket
3.0	Ammar	40.0	musik
5.0	Rezki	34.0	menari
7.0	Choiri	34.0	mendaki
NaN	Heri	NaN	gundu
NaN	Gunawan	NaN	berenang

**YUK LATIHAN !!!**

# Merge Data Frame

## Multiple Kolom Matching

**Use Case** Terdapat nama yang sama, padahal mereka orang yang berbeda

nama	kota	hobi
Dadan	Jakarta	Memasak
Dadan	Depok	Gundu
Dadan	Jambi	Sepak Bola
Firman	Bogor	Memasak
Ujang	Malang	Renang
Heru	Tegal	Renang
Andaru	Surabaya	Renang

df\_hobi

nama	kota	umur
Dadan	Depok	60
Firman	Bogor	15
Andaru	Palembang	26
Yonathan	Medan	55
Ilham	Jakarta	44
Petra	Jakarta	37

df\_umur



# Merge Data Frame

## Multiple Kolom Matching

### Use Case

Jika **menggunakan Satu kolom matching**, maka semua nama Dadan akan berumur 60 Tahun. Padahal 60 Tahun hanya Dadan-Depok

nama	kota	hobi	umur
Dadan	Jakarta	Memasak	60
Dadan	Depok	Gundu	60
Dadan	Jambi	Sepak Bola	60
Firman	Bogor	Memasak	15
Ujang	Malang	Renang	NA
Heru	Tegal	Renang	NA
Andaru	Surabaya	Renang	26

df\_hobi

nama	kota	umur
Dadan	Depok	60
Firman	Bogor	15
Andaru	Palembang	26
Yonathan	Medan	55
Ilham	Jakarta	44
Petra	Jakarta	37

df\_umur

# Merge Data Frame

## Multiple Kolom Matching

### Use Case

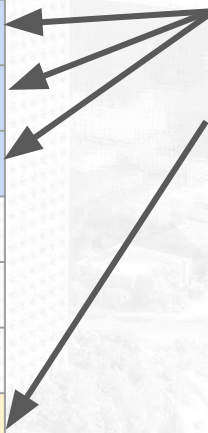
Selain itu, terjadi salah mapping untuk Andaru.  
Di kedua table memiliki kota berbeda yang menandakan orang berbeda.

nama	kota	hobi	umur
Dadan	Jakarta	Memasak	60
Dadan	Depok	Gundu	60
Dadan	Jambi	Sepak Bola	60
Firman	Bogor	Memasak	15
Ujang	Malang	Renang	NA
Heru	Tegal	Renang	NA
Andaru	Surabaya	Renang	26

df\_hobi

nama	kota	umur
Dadan	Depok	60
Firman	Bogor	15
Andaru	Palembang	26
Yonathan	Medan	55
Ilham	Jakarta	44
Petra	Jakarta	37

df\_umur



# Merge Data Frame

## Multiple Kolom Matching

### Format Syntax

```
df_1.merge(df_2,
            left_on = [col_1a,col_1b],
            right_on = [col_2a,col_2b],
            how = 'inner' **
            )
```

\*\* bisa diganti 'left' atau 'right'

df\_1

col_1a	col_1b	col
a	a	xx
b	b	xx
c	c	xx
d	d	xx
e	e	xx

df\_2

col_2a	col_2b	col
a	a	yy
c	c	yy
d	g	yy
e	f	yy
f	f	yy
g	g	yy

# **YUK LATIHAN !!!**

soal akan ditampilkan saat kelas :)

# **LATIHAN TERAKHIR!!!**

# Terima Kasih!



**M Ramadiansyah**

Machine Learning Engineer

finAcel



Muhammad Ramadiansyah

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