DigitalSkola

python

DATAFRAME II

pandas

Topik diskusi hari ini

1. *Indexing* Dataframe
2. Menghapus kolom
3. Concatenate Dataframe
4. Append Dataframe
5. *Pivoting* Dataframe
6. *Melting* Dataframe

Indexing Dataframe

Reset Index

Jika sebuah dataframe memiliki index yang repetitive, salah satu cara yang bisa ditempuh agar tidak terjadi redudansi adalah dengan me-reset index

```
data2.head()
```

	nim	nama
0	101621001	Rimala Kumar
6	101621002	Citra Tenggara
13	101621003	Rani Kumala
18	101621004	Junaedi
23	101621005	Kameela Malik

reset_index()

```
data2.reset_index()
```

	index	nim	nama
0	0	101621001	Rimala Kumar
1	6	101621002	Citra Tenggara
2	13	101621003	Rani Kumala
3	18	101621004	Junaedi
4	23	101621005	Kameela Malik

Indexing Dataframe

Set Column as Index

Jika sebuah dataframe memiliki index yang spesifik pada kolomnya, kita bisa memberikan posisi index kepada
kolom yang kita inginkan

```
data2.head()
```

	nim	nama
0	101621001	Rimala Kumar
6	101621002	Citra Tenggara
13	101621003	Rani Kumala
18	101621004	Junaedi
23	101621005	Kameela Malik

set_index()

```
data2.set_index('nim')
```

	nama
nim	
101621001	Rimala Kumar
101621002	Citra Tenggara
101621003	Rani Kumala
101621004	Junaedi
101621005	Kameela Malik

Drop Column dalam Dataframe

Drop by Column Name

Menghapus satu atau lebih kolom data

```
data1.head()
```

	nim	nama	semester	mata_kuliah	nilai
0	101621001	Rimala Kumar	6	Kecerdasan Buatan	74
1	101621001	Rimala Kumar	6	Pengolahan Citra Digital	73
2	101621001	Rimala Kumar	6	Sistem Informasi Geografis	67
3	101621001	Rimala Kumar	7	Pengantar Studi Pustaka	92
4	101621001	Rimala Kumar	7	Kolokium	92

drop()

```
data1.drop(['nilai', 'mata_kuliah'], axis=1)
```

	nim	nama	semester
0	101621001	Rimala Kumar	6
1	101621001	Rimala Kumar	6
2	101621001	Rimala Kumar	6
3	101621001	Rimala Kumar	7
4	101621001	Rimala Kumar	7

Drop Column dalam Dataframe

Drop by Index

Menghapus satu atau lebih baris data (*row*)

```
data1.head()
```

	nim	nama	semester	mata_kuliah	nilai
0	101621001	Rimala Kumar	6	Kecerdasan Buatan	74
1	101621001	Rimala Kumar	6	Pengolahan Citra Digital	73
2	101621001	Rimala Kumar	6	Sistem Informasi Geografis	67
3	101621001	Rimala Kumar	7	Pengantar Studi Pustaka	92
4	101621001	Rimala Kumar	7	Kolokium	92

drop()

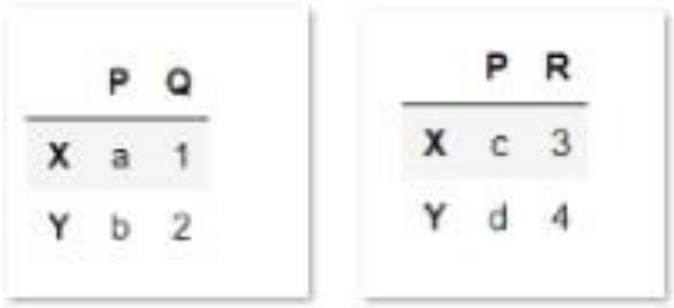
```
data1.drop(index=2, axis=0)
```

	nim	nama	semester	mata_kuliah	nilai
0	101621001	Rimala Kumar	6	Kecerdasan Buatan	74
1	101621001	Rimala Kumar	6	Pengolahan Citra Digital	73
3	101621001	Rimala Kumar	7	Pengantar Studi Pustaka	92
4	101621001	Rimala Kumar	7	Kolokium	92

Join & Merge Dataframe

Sebelumnya telah membahas **Merge**, tapi ada fungsi lain yaitu **Join**

- ✔ Pada dasarnya, konsep menggabungkan tabel dalam dataframe sama dengan SQL Query.
- ✔ Satu-satunya perbedaan adalah bagaimana kodenya.



	P	Q
X	a	1
Y	b	2

	P	R
X	c	3
Y	d	4

Join

Python3

```
joined_df = left.join(right, lsuffix='_')
print(joined_df)
```

Output :

	P_	Q	P	R
X	a	1	c	3
Y	b	2	d	4

- ✔ Metode ***join*** mengambil dua DataFrame dan menggabungkannya berdasarkan index mereka
- ✔ Jika ada nama kolom yang tumpang tindih, gabungan akan meminta untuk menambahkan sufiks ke nama kolom yang tumpang tindih.
- ✔ Dua kerangka data di atas memiliki nama kolom yang tumpang tindih , yaitu P.

Join & Merge Dataframe

	P	Q
X	a	1
Y	b	2

	P	R
X	c	3
Y	d	4

Merge

- ✔ Di sini, perhatikan bahwa metode ***merge*** menghancurkan index.
- ✔ Tapi ***merge*** lebih fleksibel
- ✔ Kita dapat menentukan kolom yang tumpang tindih dengan parameter aktif, atau dapat secara terpisah menentukannya dengan parameter left_on dan right_on.

Python3

```
merged_df = left.merge(right, on='P', how='outer')
print(merged_df)
```

Output :

	P	Q	R
0	a	1.0	NaN
1	b	2.0	NaN
2	c	NaN	3.0
3	d	NaN	4.0

Join & Merge Dataframe

	P	Q
X	a	1
Y	b	2

	P	R
X	c	3
Y	d	4

Merge

```
merged_df = left.merge(right, left_index=True,
                       right_index=True, suffixes=['_', ''])
print(merged_df)
```

Output :

	P_	Q	P	R
X	a	1	c	3
Y	b	2	d	4

- ✔ Dengan ***merge,*** kita bahkan bisa melakukan apa yang ***join*** bisa lakukan

Concatenate & Append Dataframe

- ***Concatenate*** dalam Dataframe dibuat lebih fleksibel daripada tabel dalam basis data.
- Karena kita bisa memilih concat secara horizontal (kolom) atau vertikal (baris).

```
# axis = 1 is horizontally (columns)
# axis = 0 is vertically (rows)
df = pd.concat([df1, df2], axis=1)
```

Concatenate & Append Dataframe

- ***Concatenate*** untuk menambahkan kolom

```
In [9]: result = pd.concat([df1, df4], axis=1)
```

df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df4

	B	D	F
2	B2	D2	F2
3	B3	D3	F3
6	B6	D6	F6
7	B7	D7	F7

Result

	A	B	C	D	B	D	F
0	A0	B0	C0	D0	NaN	NaN	NaN
1	A1	B1	C1	D1	NaN	NaN	NaN
2	A2	B2	C2	D2	B2	D2	F2
3	A3	B3	C3	D3	B3	D3	F3
6	NaN	NaN	NaN	NaN	B6	D6	F6
7	NaN	NaN	NaN	NaN	B7	D7	F7

Concatenate & Append Dataframe

```
In [4]: frames = [df1, df2, df3]

In [5]: result = pd.concat(frames)
```

- ✔ ***Concatenate*** untuk menambahkan baris

df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df2

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

df3

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

Result

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

Concatenate & Append Dataframe

- ✔ ***Append*** dalam Dataframe dibuat untuk menambahkan baris (*row*) baru.
- ✔ Untuk menambahkan kerangka data di kolom yang sama, nama kolom di antara dua tabel harus sama.
- ✔ Jika ada perbedaan nama kolom, kolom baru akan dibuat.

```
In [13]: result = df1.append(df2)
```

df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df2

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

Result

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

Pivoting Dataframe

- ✔ Membuat tabel pivot bergaya spreadsheet dalam bentuk Dataframe.
- ✔ Level dalam tabel pivot akan disimpan dalam bentuk MultiIndex (indeks hierarkis) pada indeks dan kolom.

	A	B	C	D	E
0	foo	one	small	1	2
1	foo	one	large	2	4
2	foo	one	large	2	5
3	foo	two	small	3	5
4	foo	two	small	3	6
5	bar	one	large	4	6
6	bar	one	small	5	8
7	bar	two	small	6	9
8	bar	two	large	7	9

```
>>> table = pd.pivot_table(df, values='D', index=['A', 'B'],
...                        columns=['C'], aggfunc=np.sum)
>>> table
C        large  small
A   B
bar one    4.0    5.0
    two    7.0    6.0
foo one    4.0    1.0
    two    NaN    6.0
```

Melting Dataframe

- ✔ Melepaskan Pivot DataFrame dari format lebar ke format panjang, secara opsional biarkan pengidentifikasi ditetapkan.

```
>>> df
   A  B  C
0  a  1  2
1  b  3  4
2  c  5  6
```

```
>>> pd.melt(df, id_vars=['A'], value_vars=['B'])
   A variable  value
0  a        B      1
1  b        B      3
2  c        B      5
```

```
>>> pd.melt(df, id_vars=['A'], value_vars=['B', 'C'])
   A variable  value
0  a        B      1
1  b        B      3
2  c        B      5
3  a        C      2
4  b        C      4
5  c        C      6
```

DigitalSkola

Thank YOU