

Nama : Fitri Nurul 'Azmi

NIM : 2100015034

Penggabungan Series/DataFrame

Terdapat beberapa metode untuk menggabungkan series/dataframe di Pandas, yaitu :

1. Append
2. Concat
3. Merge
4. Join

▼ Import Library

Import beberapa Library berikut untuk digunakan di sepanjang notebook ini :

```
import pandas as pd
```

▼ Append

Method.append() dapat digunakan pada dataframe/series yang ditunjukkan untuk menambahkan row-nya saja.

```
# Membuat series of int (s1) dan series of string (s2)
s1 = pd.Series([1,2,3,4,5,6])
s2 = pd.Series(["a","b","c","d","f"])
# Membuat metode append
s1_append_s2 = s1.append(s2)
print("Series-append:\n", s1_append_s2)
```

```
Series-append:
0    1
1    2
2    3
3    4
4    5
5    6
0    a
1    b
2    c
3    d
4    f
dtype: object
<ipython-input-6-7a9ee217af5d>:5: FutureWarning: The series.append method is deprecated and will be removed from pandas in a future
s1_append_s2 = s1.append(s2)
```

Untuk DataFrame

```
# Buat dataframe df1 dan df2
df1 = pd.DataFrame({'a':[1,2],
                    'b':[3,4]})
print("df1:\n",df1)
df2 = pd.DataFrame({'b':[1,2],
                    'a':[3,4]})
print("df2:\n",df2)
# Terapkan metode append
df1_append_df2 = df1.append(df2)
print("DataFrame-append:\n", df1_append_df2)
```

```
df1:
   a  b
0  1  3
1  2  4
df2:
   b  a
0  1  3
1  2  4
DataFrame-append:
   a  b
0  1  3
1  2  4
0  3  1
```

```

1 4 2
<ipython-input-11-709cc5c11c82>:9: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future
df1_append_df2 = df1.append(df2)

```

▼ Concat

Method.concat() dapat digunakan pada dataframe yang ditunjukkan untuk penggabungan baik dalam row-wise(adalam arah) atau column-wise

```

# Membuat data frame df1 dan df2
# Buat dataframe df1 dan df2
df1 = pd.DataFrame({'a':[1,2],
                    'b':[3,4]})

print("df1:\n",df1)
df2 = pd.DataFrame({'b':[1,2],
                    'a':[3,4]})

print("df2:\n",df2)
# Menerapkan metode concat row-wise
row_wise_concat = pd.concat ([df1,df2])
print("Row-wise-concat:\n", row_wise_concat)

```

```

df1:
   a  b
0  1  3
1  2  4
df2:
   b  a
0  1  3
1  2  4
Row-wise-concat:
   a  b
0  1  3
1  2  4
0  3  1
1  4  2

```

Untuk menerapkan concat **column-wise**

```

# Terapkan metode concat column-wise
col_wise_concat = pd.concat([df1, df2], axis = 1)
print("column-wise - concat:\n", col_wise_concat)

```

```

column-wise - concat:
   a  b  b  a
0  1  3  1  3
1  2  4  2  4

```

Dapa juga menambahkan identifier dari dataframe untuk data yang ditambahkan

```

# Penambahan identifier --> membentuk hasil menggabungkan multiindex
multiindex_concat = pd.concat([df1,df2],axis = 0, keys = ['df1','df2'])
print("Multiindex - concat:\n", multiindex_concat)

```

```

Multiindex - concat:
      a  b
df1 0  1  3
    1  2  4
df2 0  3  1
    1  4  2

```

▼ Merge

Method.merge() untuk menggabungkan series/Dataframe yang bentuknya mirip dengan syntax join di SQL, specify left and right tables, join key dan how join(left, right, inner, full, outer)

```

# Membuat dataframe df1 dan df2
df1 = pd.DataFrame({
    'key' : ['k1', 'k2', 'k3', 'k4', 'k5'],
    'val1' : [200, 500, 0, 500, 100],
    'val2' : [30, 50, 100, 20, 10]
})
print("df1:\n", df1)
df2 = pd.DataFrame({
    'key' : ['k1', 'k3', 'k5', 'k7', 'k10'],

```

```
'val3' : [1, 2, 3, 4, 5],
'val4' : [6, 7, 8, 9, 10]
})
print("df2:\n", df2)
```

```
df1:
  key  val1  val2
0  k1   200    30
1  k2   500    50
2  k3     0   100
3  k4   500    20
4  k5   100    10
df2:
  key  val3  val4
0  k1     1     6
1  k3     2     7
2  k5     3     8
3  k7     4     9
4  k10    5    10
```

pd.merge yang ekivalen dengan SQL **left join**

```
# Merge yang ekivalen dengan SQL left join
merge_df_left = pd.merge(left = df1, right = df2, how = 'left', left_on = 'key', right_on = 'key')
print('merge - left:\n', merge_df_left)
```

```
merge - left:
  key  val1  val2  val3  val4
0  k1   200    30   1.0   6.0
1  k2   500    50   NaN   NaN
2  k3     0   100   2.0   7.0
3  k4   500    20   NaN   NaN
4  k5   100    10   3.0   8.0
```

pd.merge yang ekivalen dengan SQL **right join**

```
# Merge yang ekivalen dengan SQL right join
merge_df_right = pd.merge(left = df1, right = df2, how = 'right', left_on = 'key', right_on = 'key')
print('merge - right:\n', merge_df_right)
```

```
merge - right:
  key  val1  val2  val3  val4
0  k1  200.0  30.0    1     6
1  k3    0.0  100.0    2     7
2  k5  100.0   10.0    3     8
3  k7    NaN   NaN    4     9
4  k10   NaN   NaN    5    10
```

pd.merge yang ekivalen dengan SQL **inner join**

```
# Merge yang ekivalen dengan SQL inner join
merge_df_inner = pd.merge(left = df1, right = df2, how = 'inner', left_on = 'key', right_on = 'key')
print('merge - innert:\n', merge_df_inner)
```

```
merge - innert:
  key  val1  val2  val3  val4
0  k1   200    30     1     6
1  k3     0   100     2     7
2  k5   100    10     3     8
```

pd.merge yang ekivalen dengan SQL **outer join**

```
# Merge yang ekivalen dengan SQL outer join
merge_df_outer = pd.merge(left = df1, right = df2, how = 'outer', left_on = 'key', right_on = 'key')
print('merge - outer:\n', merge_df_outer)
```

```
merge - outer:
  key  val1  val2  val3  val4
0  k1  200.0  30.0   1.0   6.0
1  k2  500.0  50.0   NaN   NaN
2  k3    0.0  100.0   2.0   7.0
3  k4  500.0  20.0   NaN   NaN
4  k5  100.0  10.0   3.0   8.0
5  k7    NaN   NaN   4.0   9.0
6  k10   NaN   NaN   5.0  10.0
```

Bagaimana jika salah satu dataframe atau keseluruhan dataframe yang akan digabungkan tersebut memiliki multiindex?

```
# Membuat dataframe df1 dan df2
df1 = pd.DataFrame({
    'key' : ['k1', 'k2', 'k3', 'k4', 'k5'],
    'val1' : [200, 500, 0, 500, 100],
    'val2' : [30, 50, 100, 20, 10]
}).set_index(['key', 'val2'])
print("df1:\n", df1)
df2 = pd.DataFrame({
    'key' : ['k1', 'k3', 'k5', 'k7', 'k10'],
    'val3' : [1, 2, 3, 4, 5],
    'val4' : [6, 7, 8, 9, 10]
}).set_index(['key', 'val3'])
print("df2:\n", df2)
```

```
df1:
      key val2  val1
k1  k1    30   200
k2  k2    50   500
k3  k3   100     0
k4  k4    20   500
k5  k5    10   100
df2:
      key val3  val4
k1  k1     1     6
k3  k3     2     7
k5  k5     3     8
k7  k7     4     9
k10 k10    5    10
```

Jika digunakan secara langsung seperti yang telah dilakukan pada bagian sebelumnya

```
# Merge dataframe yang memiliki multi index
# df_merge = pd.merge(df1,df2)
# print('Merging dataframe:\n', df_merge)

# Merge dataframe yang memiliki multi index
df_merge = pd.merge(df1.reset_index(), df2.reset_index())
print('Merging dataframe:\n', df_merge)
```

```
Merging dataframe:
   key  val2  val1  val3  val4
0  k1    30   200     1     6
1  k3   100     0     2     7
2  k5    10   100     3     8
```

▼ Join

Method **join()** digunakan pada dataframe untuk menggabungkan kedua data dengan set index pada kedua data dengan det index pada tabel tersebut sebagai join key, tanpa index, hal ini tidak akan berhasil.

```
# Buat dataframe df1 dan df2
df1 = pd.DataFrame({
    'key' : ['k1', 'k2', 'k3', 'k4', 'k5'],
    'val1' : [200, 500, 0, 500, 100],
    'val2' : [30, 50, 100, 20, 10]
})
df2 = pd.DataFrame({
    'key' : ['k1', 'k3', 'k5', 'k7', 'k10'],
    'val3' : [1, 2, 3, 4, 5],
    'val4' : [6, 7, 8, 9, 10]
})
# Penerapan join dengan menggunakan set_index dan keyword how
# df1.join(df2) # <-- akan error

print(df1.set_index('key').join(df2.set_index('key')))
```

```
      val1  val2  val3  val4
key
k1    200    30    1.0    6.0
k2    500    50    NaN    NaN
k3     0   100    2.0    7.0
k4    500    20    NaN    NaN
k5    100    10    3.0    8.0
```

Untuk tipe join yang lain(contoh = inner), harus menspecify keyword how = 'inner' seperti yang dicontohkan berikut ini

```
join_df = df1.set_index('key').join(df2.set_index('key'), how = 'inner')
print(join_df)
```

	val1	val2	val3	val4
key				
k1	200	30	1	6
k3	0	100	2	7
k5	100	10	3	8

✓ 0s completed at 7:23 PM

×