

# Quiz Pekan - 2

Isikan dengan email yang terdaftar pada [sanbercode.com](https://sanbercode.com)

Jangan lupa berdoa dan berbuatlah jujur ٩(ง ~~~)—☆°. \*°.° (,,~~,,)

\* Required

1. Email address \*

---

## Quiz Pekan - 2

2. Mana attribute dibawah ini yang bisa di gunakan untuk mendapatkan numpy array dari suatu column pada DataFrame? \* 1 point

*Mark only one oval.*

☐ .values()

☒ .values

☐ .array()

☐ .array

☐ np.array(DataFrame)

3. Diketahui sebuah DataFrame 'df' sebagai berikut, perintah untuk menghapus row dengan Country\_Region berupa United Kingdom adalah? \* 1 point

	OBJECTID	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active
0	18	US	1593185632000	40.000000	-100.000000	2425814	124509	663562.0	1561508
1	22	Brazil	1593185632000	-14.235000	-51.925300	1228114	54971	679524.0	493619
2	14	Russia	1593185632000	61.524000	105.318800	619936	8770	383524.0	227642
3	27	India	1593185632000	20.593684	78.962880	490401	15301	285637.0	189463
4	17	United Kingdom	1593185632000	55.000000	-3.000000	310836	43498	1361.0	265977
5	23	Peru	1593185632000	-9.190000	-75.015200	268602	8761	151225.0	108616
6	21	Chile	1593185632000	-35.675100	-71.543000	259064	4903	219327.0	34834
7	19	Spain	1593185632000	40.463667	-3.749220	247905	28338	150376.0	69192
8	11	Italy	1593185632000	41.871900	12.567400	239706	34678	186725.0	18303
9	97	Iran	1593185632000	32.427908	53.688046	217724	10239	177852.0	29633

Mark only one oval.

- ☒ df.drop(4)
- ☐ df.pop(17)
- ☐ df.drop(17)
- ☐ df.remove(df['Country\_Region'] == 'United Kingdom')
- ☐ df.remove(4)

4. Mana method berikut yang akan menghasilkan summary statistic dari DataFrame? \* 1 point

Mark only one oval.

- ☐ .summary()
- ☐ .info()
- ☒ .describe()
- ☐ .stats()
- ☐ .statistics()

5. Gambar berikut adalah hasil groupby DataFrame 'df' pada kolom 'buyer\_userid'. Manakah perintah yang TIDAK dapat digunakan untuk mengetahui nilai 'message\_id' dari 'buyer\_userid' dengan ID 'B1001836366'?  
~~ada 2 jawaban betul~~ [NOTE: SOAL INI DI ANULIR DAN TIDAK AKAN DIMASUKKAN KE DALAM PERHITUNGAN] \*

0 points

	message_id	is_buyer	is_seller
buyer_userid			
B1000086632	15193.0	0.0	1.0
B1000431826	15205.5	0.0	1.0
B1000486824	15216.5	0.5	0.5
B1000520503	15233.5	1.0	0.0
B1000844336	15249.5	0.0	1.0
B1000915829	15254.5	1.0	0.0
B1000986701	347.5	1.0	0.0
B1001005156	15163.5	1.0	0.0
B1001037055	4199.5	0.5	0.5
B1001224886	15277.5	0.0	1.0
B1001836366	15286.5	0.0	1.0
B1002042304	351.0	1.0	0.0
B1002142622	15290.0	1.0	0.0
B1002326800	15295.5	0.0	1.0
B1002674723	15302.0	1.0	0.0

Check all that apply.

- ☐ df.loc['B1001836366'].loc['message\_id']
- ☒ df['B1001836366']['message\_id']
- ☐ df.iloc[10].iloc[0]
- ☐ df['message\_id']['B1001836366']
- ☐ df.loc['B1001836366'].message\_id

6. Diketahui sebuah DataFrame 'df' seperti pada gambar berikut. Jika nilai 'price' berada di atas 8 maka termasuk mahal. Manakah perintah yang benar untuk menampilkan 'df' dengan nilai 'price' yang tidak mahal? #pilih 3 jawaban \*
- 1 point

	fruit	price	quantity
0	Apple	4	10
1	Pear	7	14
2	Durian	10	2

Check all that apply.

- ☒ `df[~(df['price'] > 8)]`
- ☒ `df[(df['price'] <= 8)]`
- ☒ `df[df.price <= 8]`
- ☐ `df.loc[df['price']>=8, :]`
- ☐ `df.loc[df.price>=8, :]`

7. Pada pandas selections berdasarkan posisi, kita dapat menggunakan metode .iloc pada DataFrame 'df' seperti berikut: `df.iloc[A, B]`. Maka nilai A dapat berupa? #pilih 3 jawaban \*
- 1 point

Check all that apply.

- ☒ list integer dari row
- ☒ integer tunggal
- ☐ boolean
- ☐ list string dari row
- ☒ slice dari row

8. Diketahui df3 merupakan DataFrame hasil gabung antara 'df1' dan 'df2' seperti pada gambar. Manakah perintah yang digunakan untuk menghasilkan 'df3'? \*
- 1 point

df1				df2		
	fruit	price	quantity		fruit	weight
0	Apple	4	10	0	Apple	10
1	Pear	7	14	1	Durian	20
2	Durian	10	2	2	Grape	2

  

df3				
	fruit	price	quantity	weight
0	Apple	4.0	10.0	10.0
1	Pear	7.0	14.0	NaN
2	Durian	10.0	2.0	20.0
3	Grape	NaN	NaN	2.0

Mark only one oval.

- ☒ df3 = pd.merge(df1, df2, on='fruit', how='outer')
- ☐ df3 = pd.merge(df1, df2, on='fruit', how='inner')
- ☐ df3 = pd.concat(df1, df2, on='fruit', how='inner')
- ☐ df3 = pd.join(df1, df2, on='fruit', how='inner')
- ☐ df3 = pd.join(df1, df2, on='fruit', how='outer')

9. Diketahui DataFrame 'df' seperti pada gambar berikut. Jike kita menggunakan perintah `pd.pivot_table(df, values='D', index=['B', 'A'], columns=['C'], aggfunc='sum')` maka output yang dihasilkan adalah? \*

1 point

	A	B	C	D	E
0	koo	Na	masaka	1	2
1	koo	Na	orraa	2	4
2	koo	Na	orraa	2	5
3	koo	Ni	masaka	3	5
4	koo	Ni	masaka	3	6
5	ree	Na	orraa	4	6
6	ree	Na	masaka	5	8
7	ree	Ni	masaka	6	9
8	ree	Ni	orraa	7	9

Mark only one oval.

	C	masaka	orraa
A	B		
koo	Na	1.0	4.0
	Ni	6.0	NaN
ree	Na	5.0	4.0
	Ni	6.0	7.0

☐ (~~)

	C	masaka	orraa
B	A		
Na	koo	1.0	4.0
	ree	5.0	4.0
Ni	koo	6.0	NaN
	ree	6.0	7.0

☒ (~~)

	C	masaka	orraa
B	A		
Na	koo	2.0	4.5
	ree	8.0	6.0
Ni	koo	5.5	NaN
	ree	9.0	9.0

☐ (,,~~,,)

	C	masaka	orraa
A	B		
koo	Na	2.0	5.0
	Ni	6.0	NaN
ree	Na	8.0	6.0
	Ni	9.0	9.0

☐ (~~~\*)

		C	masaka	orraa
A	B			
koo	Na	1.0	2.0	
	Ni	3.0	NaN	
ree	Na	5.0	4.0	
	Ni	6.0	7.0	

☐ ( ◦ u ◦ u ◦ )

10. Untuk menambahkan kolom baru bernama 'Baru' berisi Series kedalam DataFrame bernama 'df' pada kolom terakhir, maka perintah yang benar adalah? #pilih 3 jawaban \*

1 point

*Check all that apply.*

- ☒ df['Baru'] = Series
- ☐ df.insert('Baru') = Series
- ☒ df = df.assign(Baru = Series)
- ☒ df.insert(loc=len(df.columns), column='Baru', value=Series)
- ☐ df.loc['Baru'] = Series

11. Diketahui sebuah DataFrame dengan nama 'df', untuk mengubah menampilkan 'df1' dari 'df', maka perintah yang dapat dilakukan adalah?  
#pilih 2 jawaban \*

1 point

df			
	makanan	rating	harga
0	rendang	0.633101	2
1	sate	0.927708	11
2	bakso	0.504761	6
3	rendang	0.888030	9
4	sate	0.573661	2
5	bakso	0.955192	12
6	rendang	0.129832	1
7	sate	0.377487	8
8	bakso	0.864831	9
df1			
	makanan	rating	harga
0	bakso	0.774928	9
1	rendang	0.550321	4
2	sate	0.626285	7

Check all that apply.

- ☒ df.groupby('makanan', as\_index=False).mean()
- ☒ df.groupby('makanan').mean().reset\_index()
- ☐ df.groupby('makanan').mean()
- ☐ df.groupby('makanan').agg('mean')
- ☐ df.groupby(['makanan'], as\_index=False).agg(['mean'])



12. Untuk menghitung frekuensi masing-masing value pada suatu DataFrame 'df' maka dapat menggunakan? \* 1 point

*Mark only one oval.*

- ☒ pd.value\_counts(df.values.ravel())
- ☐ pd.count(df)
- ☐ df.value\_counts()
- ☐ pd.ravel(df.values\_counts())
- ☐ df.count(df.ravel())

13. Manakah perintah berikut yang dapat digunakan untuk menukar antara row index 1 dan row index 2 pada DataFrame 'df'? \* 1 point

*Mark only one oval.*

- ☒ df.iloc[1], df.iloc[2] = df.iloc[2], df.iloc[1].copy(deep = True)
- ☐ df.iloc[1], df.iloc[2] = df.iloc[2], df.iloc[1] #Tidak dapat digunakan pada dataframe dari numpy array
- ☐ df.swap(df.row[1], df.row[2])
- ☐ df.swap(df.row[1], df.row[2], inplace=True)
- ☐ df.iloc[1], df.iloc[2] = df.swap(df.iloc[2], df.iloc[1])

14. Untuk menampilkan DataFrame 'df1' dari DataFrame 'df' seperti pada gambar, maka perintah yang dapat digunakan adalah? ~~#pilih 3 jawaban\*~~

1 point

df						df1					
0	1	2	3	4		0	1	2	3	4	
0	0	1	2	3	4	4	20	21	22	23	24
1	5	6	7	8	9	3	5	6	7	8	9
2	10	11	12	13	14	2	10	11	12	13	14
3	15	16	17	18	19	1	5	6	7	8	9
4	20	21	22	23	24	0	0	1	2	3	4

Check all that apply.

- ☒ df.iloc[df.index.to\_list()[::-1]]
- ☒ df.iloc[::-1, :]
- ☒ df.loc[df.index[::-1], :]
- ☒ df.iloc[::-1, df.index.to\_list()]
- ☒ df.loc[df.index[::-1], df.index[::-1]]

15. Pilih semua opsi yang dapat dijadikan DataFrame \*

1 point

Check all that apply.

- ☒ Series
- ☒ list
- ☒ numpy array
- ☒ dictionary
- ☒ DataFrame

16. Pilih semua opsi yang termasuk fungsi reshaping suatu DataFrame \*

1 point

*Check all that apply.*

- ☒ melt
- ☒ pivot\_table
- ☒ pivot
- ☒ stack
- ☒ unstack

---

This content is neither created nor endorsed by Google.

Google Forms