Data Noise, Manipulasi Data, Normalisasi Data		
Praktikan	**Aslab**	
Nama: xxxx Nim: xxxx	Annur Hangga Prihadi	065001800028
	Faiz Kumara	065001800003

PRAKTIKUM 2

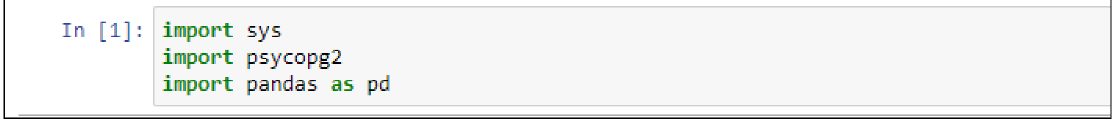
DATA SAINS DAN ANALITIK

Topik pertemuan praktikum kedua adalah mengetahui cara menemukan data missing bersifat noise dan membersihkan data noise tersebut, manipulasi data berdasarkan grup, manipulasi data berdasarkan filter, normalisasi Zscore dan Minmax.

Latihan 1

1. Memasang library yang dibutuhkan

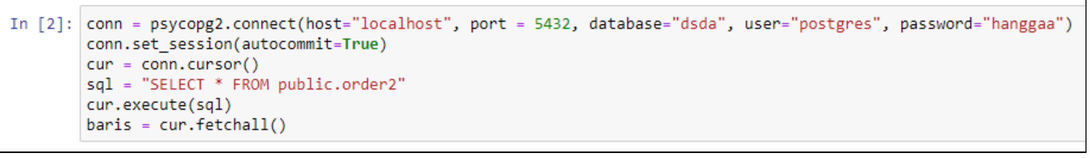
```
import sys
import psycopg2
import pandas as pd
```



```
In [1]: import sys
        import psycopg2
        import pandas as pd
```

2. Koneksikan ke database

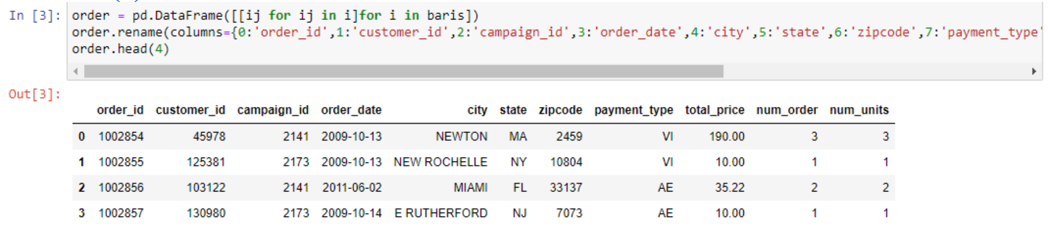
```
conn = psycopg2.connect(host="localhost", port = XXXX, database="dsda", user="postgres", password="XXXX")
conn.set_session(autocommit=True)
cur = conn.cursor()
sql = "SELECT * FROM public.NAMA_TABEL"
cur.execute(sql)
baris = cur.fetchall()
```



```
In [2]: conn = psycopg2.connect(host="localhost", port = 5432, database="dsda", user="postgres", password="hanggaa")
        conn.set_session(autocommit=True)
        cur = conn.cursor()
        sql = "SELECT * FROM public.order2"
        cur.execute(sql)
        baris = cur.fetchall()
```

3. Listing raw dataset, rename, dan menampilkan n data

```
order = pd.DataFrame([[ij for ij in i]for i in baris])
order.rename(columns={0:'order_id',1:'customer_id',2:'campaign_id',3:'order_date',4:'city',5:'state',6:
'zipcode',7:'payment_type',8:'total_price',9:'num_order',10:'num_units'},inplace=True)
order.head(4)
```



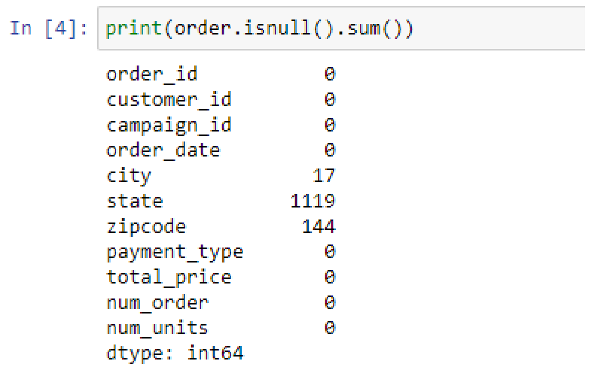
```
In [3]: order = pd.DataFrame([[ij for ij in i]for i in baris])
        order.rename(columns={0:'order_id',1:'customer_id',2:'campaign_id',3:'order_date',4:'city',5:'state',6:'zipcode',7:'payment_type'
        order.head(4)
```

Out[3]:

	order_id	customer_id	campaign_id	order_date	city	state	zipcode	payment_type	total_price	num_order	num_units
0	1002854	45978	2141	2009-10-13	NEWTON	MA	2459	VI	190.00	3	3
1	1002855	125381	2173	2009-10-13	NEW ROCHELLE	NY	10804	VI	10.00	1	1
2	1002856	103122	2141	2011-06-02	MIAMI	FL	33137	AE	35.22	2	2
3	1002857	130980	2173	2009-10-14	E RUTHERFORD	NJ	7073	AE	10.00	1	1

4. Cek missing value/null

print(order.isnull().sum())

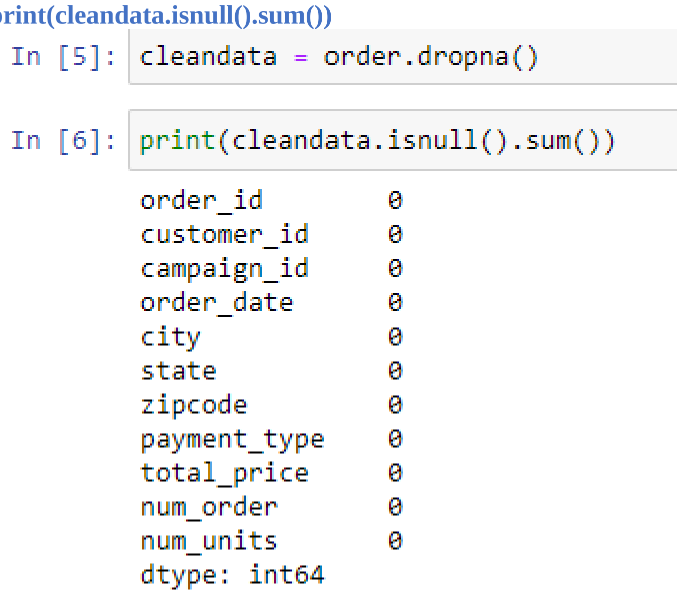


```
In [4]: print(order.isnull().sum())

        order_id          0
        customer_id       0
        campaign_id       0
        order_date        0
        city             17
        state          1119
        zipcode         144
        payment_type      0
        total_price       0
        num_order         0
        num_units         0
        dtype: int64
```

5. Fix missing value/null dan cek missing value/null setelah dibersihkan

cleandata = order.dropna()
print(cleandata.isnull().sum())



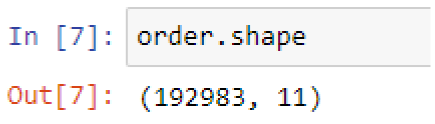
```
In [5]: cleandata = order.dropna()

In [6]: print(cleandata.isnull().sum())

        order_id        0
        customer_id     0
        campaign_id     0
        order_date      0
        city            0
        state           0
        zipcode         0
        payment_type    0
        total_price     0
        num_order       0
        num_units       0
        dtype: int64
```

6. Cek dimensi raw data

order.shape

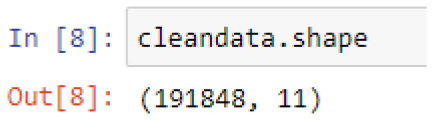


```
In [7]: order.shape

Out[7]: (192983, 11)
```

7. Cek dimensi data yang sudah bersih

cleandata.shape

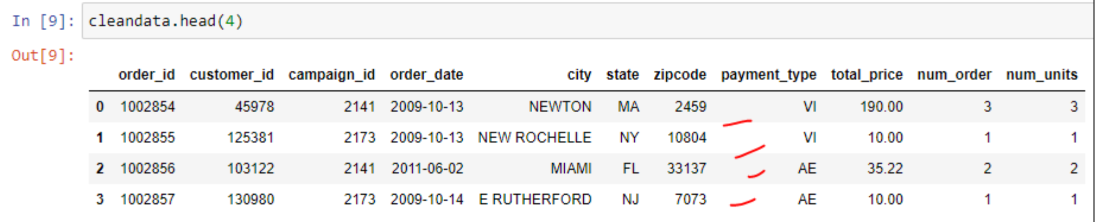


```
In [8]: cleandata.shape

Out[8]: (191848, 11)
```

8. Tampilkan n data yang sudah bersih

cleandata.head(4)

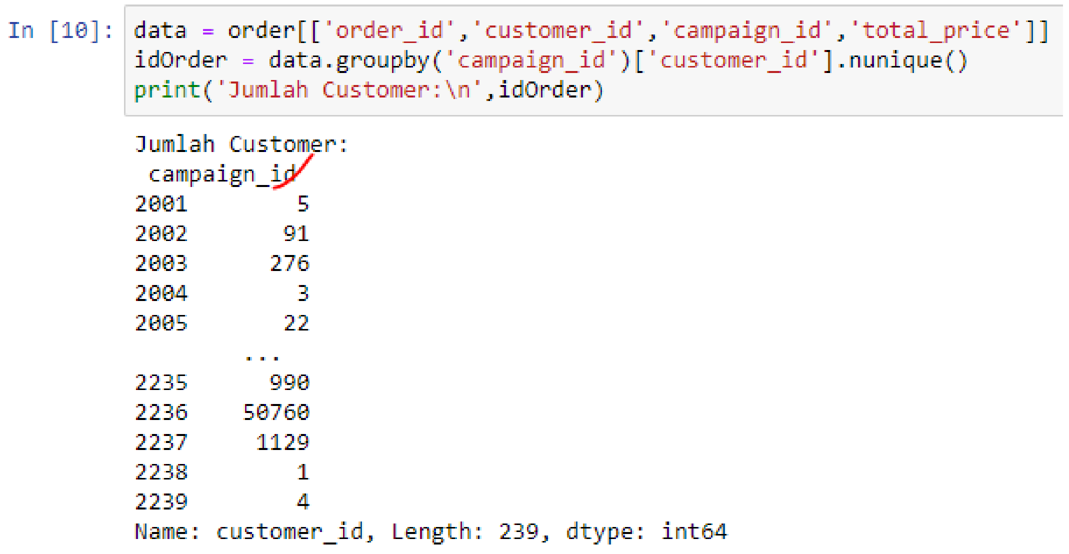


```
In [9]: cleandata.head(4)
Out[9]:
```

	order_id	customer_id	campaign_id	order_date	city	state	zipcode	payment_type	total_price	num_order	num_units
0	1002854	45978	2141	2009-10-13	NEWTON	MA	2459	VI	190.00	3	3
1	1002855	125381	2173	2009-10-13	NEW ROCHELLE	NY	10804	VI	10.00	1	1
2	1002856	103122	2141	2011-06-02	MIAMI	FL	33137	AE	35.22	2	2
3	1002857	130980	2173	2009-10-14	E RUTHERFORD	NJ	7073	AE	10.00	1	1

9. Manipulasi berdasarkan grup data

data = order[['order_id','customer_id','campaign_id','total_price']]
idOrder = data.groupby('campaign_id')['customer_id'].nunique()
print('Jumlah Customer:\n',idOrder)

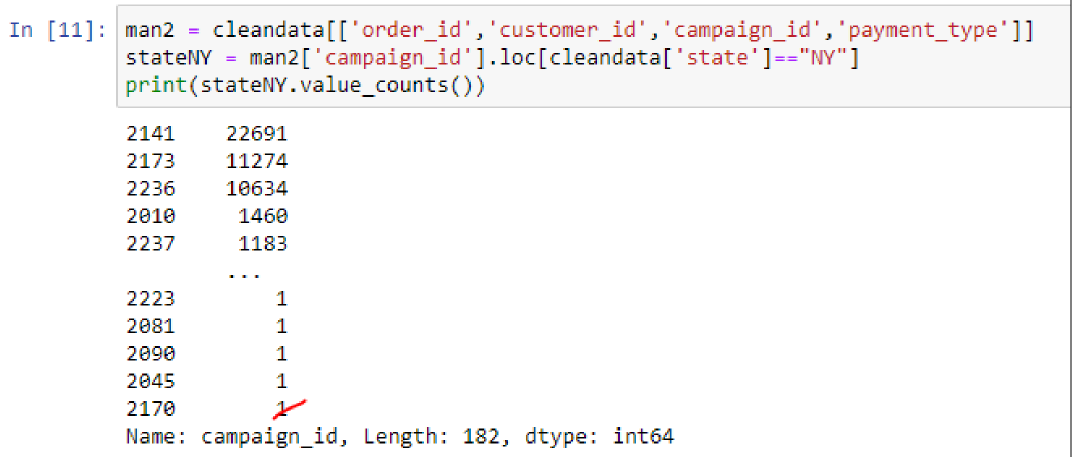


```
In [10]: data = order[['order_id','customer_id','campaign_id','total_price']]
         idOrder = data.groupby('campaign_id')['customer_id'].nunique()
         print('Jumlah Customer:\n',idOrder)

         Jumlah Customer:
          campaign_id
         2001        5
         2002       91
         2003      276
         2004        3
         2005       22
                 ...
         2235      990
         2236    50760
         2237     1129
         2238        1
         2239        4
         Name: customer_id, Length: 239, dtype: int64
```

10. Manipulasi berdasarkan filter data NY

man2 = cleandata[['order_id','customer_id','campaign_id','payment_type']]
stateNY = man2['campaign_id'].loc[cleandata['state']=="NY"]
print(stateNY.value_counts())

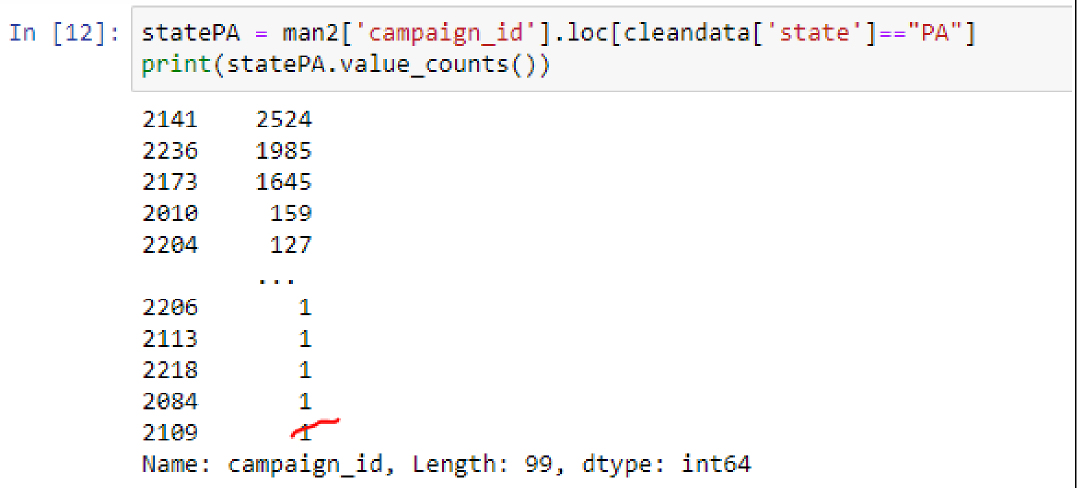


```
In [11]: man2 = cleandata[['order_id','customer_id','campaign_id','payment_type']]
         stateNY = man2['campaign_id'].loc[cleandata['state']=="NY"]
         print(stateNY.value_counts())

         2141    22691
         2173    11274
         2236    10634
         2010     1460
         2237     1183
                 ...
         2223        1
         2081        1
         2090        1
         2045        1
         2170        1
         Name: campaign_id, Length: 182, dtype: int64
```

11. Manipulasi berdasarkan filter data PA

```
statePA = man2['campaign_id'].loc[cleandata['state']=="PA"]
print(statePA.value_counts())
```

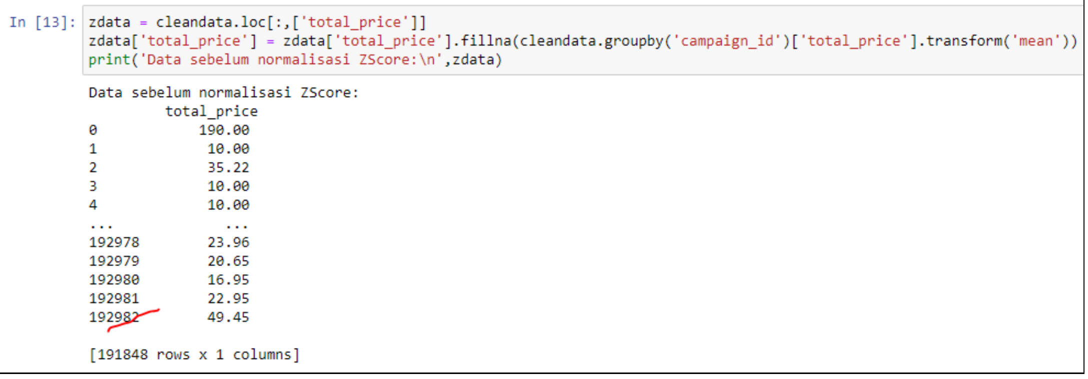


```
In [12]: statePA = man2['campaign_id'].loc[cleandata['state']=="PA"]
         print(statePA.value_counts())

         2141    2524
         2236    1985
         2173    1645
         2010     159
         2204     127
                 ...
         2206       1
         2113       1
         2218       1
         2084       1
         2109       1
         Name: campaign_id, Length: 99, dtype: int64
```

12. Normalisasi Zscore (Sebelum)

```
zdata = cleandata.loc[:,['total_price']]
zdata['total_price'] =
zdata['total_price'].fillna(cleandata.groupby('campaign_id')['total_price'].transform('mean'))
print('Data sebelum normalisasi ZScore:\n',zdata)
```



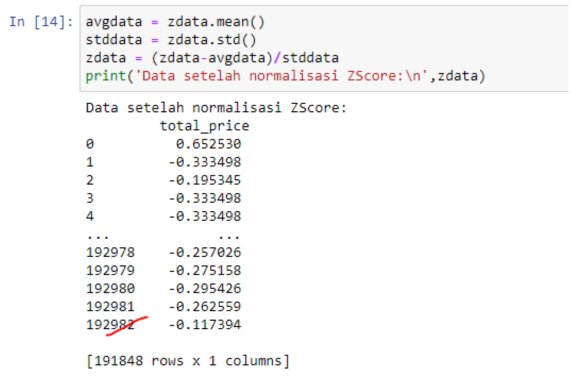
```
In [13]: zdata = cleandata.loc[:,['total_price']]
         zdata['total_price'] = zdata['total_price'].fillna(cleandata.groupby('campaign_id')['total_price'].transform('mean'))
         print('Data sebelum normalisasi ZScore:\n',zdata)

         Data sebelum normalisasi ZScore:
                  total_price
         0             190.00
         1              10.00
         2              35.22
         3              10.00
         4              10.00
         ...              ...
         192978         23.96
         192979         20.65
         192980         16.95
         192981         22.95
         192982         49.45

         [191848 rows x 1 columns]
```

13. Normalisasi Zscore (Sesudah)

```
avgdata = zdata.mean()
stddata = zdata.std()
zdata = (zdata-avgdata)/stddata
print('Data setelah normalisasi ZScore:\n',zdata)
```



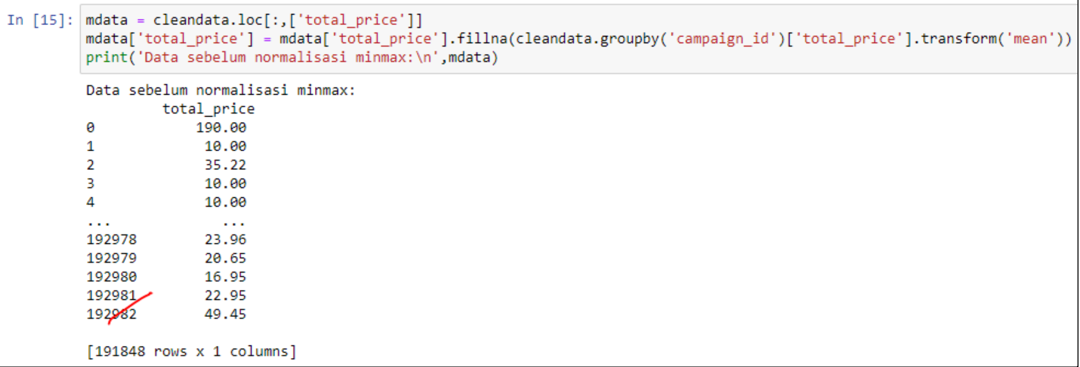
```
In [14]: avgdata = zdata.mean()
         stddata = zdata.std()
         zdata = (zdata-avgdata)/stddata
         print('Data setelah normalisasi ZScore:\n',zdata)

         Data setelah normalisasi ZScore:
                  total_price
         0           0.652530
         1          -0.333498
         2          -0.195345
         3          -0.333498
         4          -0.333498
         ...              ...
         192978     -0.257026
         192979     -0.275158
         192980     -0.295426
         192981     -0.262559
         192982     -0.117394

         [191848 rows x 1 columns]
```

14. Normalisasi Minmax (Sebelum)

```
mdata = cleandata.loc[:,['total_price']]
mdata['total_price'] =
mdata['total_price'].fillna(cleandata.groupby('campaign_id')['total_price'].transform('mean'))
print('Data sebelum normalisasi minmax:\n',mdata)
```



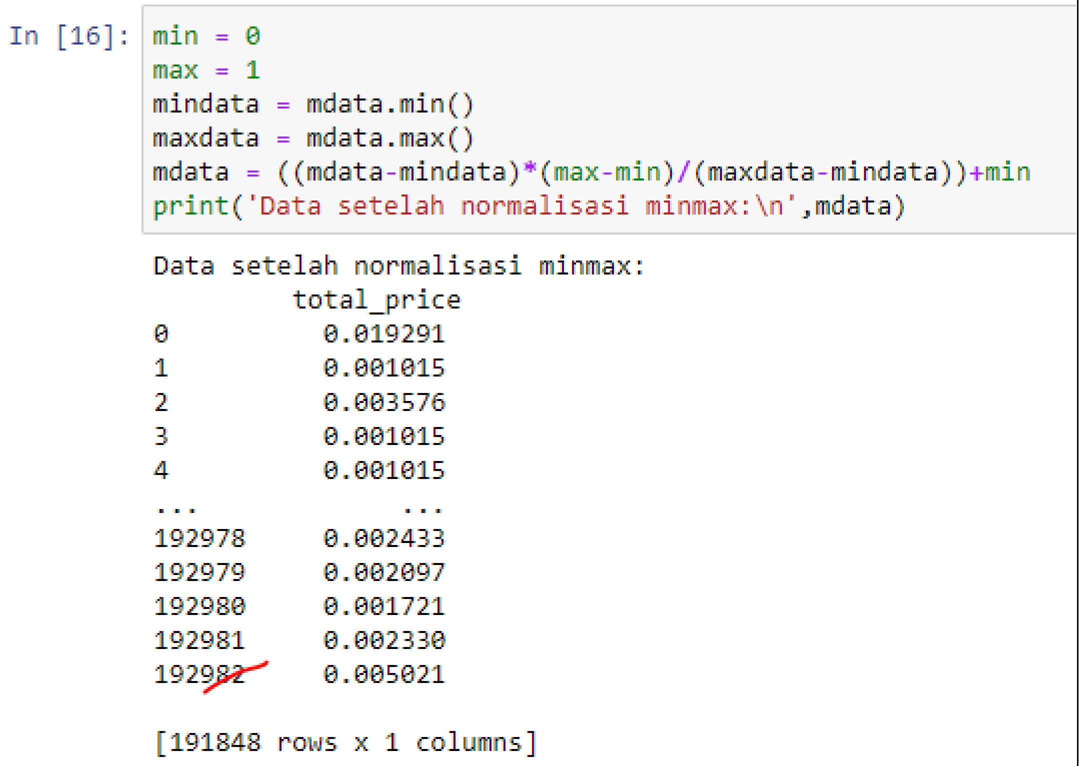
```
In [15]: mdata = cleandata.loc[:,['total_price']]
         mdata['total_price'] = mdata['total_price'].fillna(cleandata.groupby('campaign_id')['total_price'].transform('mean'))
         print('Data sebelum normalisasi minmax:\n',mdata)

         Data sebelum normalisasi minmax:
                   total_price
         0             190.00
         1              10.00
         2              35.22
         3              10.00
         4              10.00
         ...              ...
         192978         23.96
         192979         20.65
         192980         16.95
         192981         22.95
         192982         49.45

         [191848 rows x 1 columns]
```

15. Normalisasi Minmax (Sesudah)

```
min = 0
max = 1
mindata = mdata.min()
maxdata = mdata.max()
mdata = ((mdata-mindata)*(max-min)/(maxdata-mindata))+min
print('Data setelah normalisasi minmax:\n',mdata)
```



```
In [16]: min = 0
         max = 1
         mindata = mdata.min()
         maxdata = mdata.max()
         mdata = ((mdata-mindata)*(max-min)/(maxdata-mindata))+min
         print('Data setelah normalisasi minmax:\n',mdata)

         Data setelah normalisasi minmax:
                   total_price
         0            0.019291
         1            0.001015
         2            0.003576
         3            0.001015
         4            0.001015
         ...               ...
         192978       0.002433
         192979       0.002097
         192980       0.001721
         192981       0.002330
         192982       0.005021

         [191848 rows x 1 columns]
```

Latihan 2

1. Manipulasikan data berdasarkan grup data dari kolom payment_type untuk customer_id!
2. Manipulasikan data berdasarkan grup data dari kolom city untuk customer_id!
3. Manipulasikan data berdasarkan filter data dari kolom num_units untuk kolom state yang dimana berisi value TX!
4. Manipulasikan data berdasarkan filter data dari kolom num_order untuk kolom state yang dimana berisi value FL!

Lampiran Screenshot hasil 1, 2, 3, dan 4

Input screenshot disini

Makna dari masing-masing hasil di atas!

Ketik makna disini