

I. Import Modul

Melakukan import modul pandas, numpy dan juga matplotlib. Chained assignment pada pandas dinonaktifkan dan format pandas secara global untuk angka diubah menjadi `[:,.2f]`

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import matplotlib as mpl
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from jupyterthemes import jtplot

# menonaktifkan chained assignment pada pandas
pd.options.mode.chained_assignment = None # default='warn'
pd.options.display.float_format = '{:,.2f}'.format

# setting dpi matplotlib
mpl.rcParams['figure.dpi'] = 150
%config InlineBackend.figure_format = 'retina'

# setting tema jupyter untuk plotting
jtplot.style(theme='monokai')

# setting formatter untuk floating tanpa koma di belakang
format_float0 = "{:,.0f}".format

# setting periode berdasar data
periode = "2020 - 2022 YTD"
```

II. Membaca dan merubah format data pada DataFrame

Data yang digunakan dalam analisa ini adalah data penjualan PT Panatrade Caraka 2020 - 2022 (cut off bulan Maret 2022).

```
In [2]: data = pd.concat(map(pd.read_csv, ['Sales2022.csv', 'Sales2021.csv', 'Sales2020.csv']))
data.drop(['%'], axis=1)
data
```

c:\users\johan\appdata\local\programs\python\python37\lib\site-packages\pandas\core\reshape\concat.py:304: DtypeWarning: Columns (3) have mixed types.Specify dtype option on import or set low_memory=False.

```
Out[2]:
```

	Art	Desc.	Category	Quantity	Bruto	Net	Cogs	Profit
0	P10737-41	DERBY-BLACK/ORANGE/WHITE	FWEAR	6	2,278,800.00	1,595,160.00	1,237,968.00	357,192.00
1	P30179-43	NEOWAVE-BLACK/PHANTOM/WHITE	FWEAR	3	1,049,400.00	734,580.00	542,407.68	192,172.32

	2	P30200-43	VX-WAVE-BLACK/WHITE	FWEAR	4	1,399,200.00	979,440.00	676,760.36	302,679.64
	3	P20824-40	CITY RACE-NAVY/GREY/WHITE	FWEAR	2	759,600.00	531,720.00	380,929.44	150,790.56
	4	P30178-45	NEOWAVE-OLIVE/BLACK/WHITE	FWEAR	2	699,600.00	489,720.00	361,605.12	128,114.88

	142398	901845-XL	DUPA TS - DARK NAVY/WHITE	AOTHR	1	119,800.00	56,138.28	12,529.02	43,609.26
	142399	903403-NS	SPECS STRING BAG - RED/WHITE	HBAGS	1	59,800.00	28,022.28	16,309.09	11,713.19
	142400	902523-NS	OPTIMUS BIBS - ELECTRICITY GREEN/NAVY BLUE	AOTHR	1	89,800.00	42,080.28	10,186.55	31,893.73
	142401	902523-NS	OPTIMUS BIBS - ELECTRICITY GREEN/NAVY BLUE	AOTHR	7	628,600.00	294,561.96	71,305.85	223,256.11
	142402	902523-NS	OPTIMUS BIBS - ELECTRICITY GREEN/NAVY BLUE	AOTHR	1	89,800.00	42,080.28	10,186.55	31,893.73

313716 rows × 20 columns

```
In [3]: # Merubah dtype beberapa kolom dari object menjadi float64
data.Quantity = data.Quantity.str.replace(',','').astype(float)
data.Bruto = data.Bruto.str.replace(',','').astype(float)
data.Net = data.Net.str.replace(',','').astype(float)
data.Cogs = data.Cogs.str.replace(',','').astype(float)
data.Profit = data.Profit.str.replace(',','').astype(float)
data.dtypes
```

```
Out[3]: Art                object
Desc.                  object
Category              object
Quantity             float64
Bruto                float64
Net                  float64
Cogs                 float64
Profit               float64
%                    object
season               int64
brand                object
categ                object
loc                  object
customer             object
```

```
sportcast      object
inv             object
Id Cust        object
period         int64
month          object
date           object
dtype: object
```

Berdasarkan data yang diberikan, masih terdapat redundancy pada customer name ('customer') untuk customer code ('Id Cust'). Hal ini dapat dilihat dari jumlah unique 'customer' pada beberapa 'Id Cust' yang nilainya lebih dari 1 seperti dapat dilihat di bawah. Maka dengan demikian beberapa grouping yang akan menggunakan irisan customer akan lebih terkonsentrasi menggunakan 'Id Cust' ketimbang 'customer'.

```
In [4]: data.nunique()
```

```
Out[4]: Art          14914
Desc.         2914
Category       16
Quantity       504
Bruto          5051
Net           14502
Cogs          37633
Profit        69003
%             269
season         12
brand          6
categ          16
loc            20
customer       770
sportcast      22
inv           37694
Id Cust        348
period         12
month          12
date           470
dtype: int64
```

```
In [5]: cust_map = data[['Id Cust', 'customer']].groupby(['Id Cust']).nunique()
with pd.option_context('display.max_rows', None):
    print(cust_map)
```

```
          customer
Id Cust
1.01E+07         1
1.01E+08         1
1.02E+11         1
1.04E+04         1
1.04E+07         1
101A003          1
101A005          4
101A024          2
101A026          1
101B001          1
101B003          2
101B005          1
101B010          5
101B013          2
101B014          4
101B020          3
101B021          1
101B022          2
101B023          2
101B024          1
101C006          4
```

101C008	1
101C011	1
101C034	1
101D003	3
101D010	4
101D012	2
101E005	1
101F002	1
101F003	7
101F007	1
101F008	2
101F013	1
101F015	1
101G003	1
101G012	6
101G016	1
101G017	1
101G018	1
101G019	2
101H010	1
101H011	1
101I007	1
101I011	1
101I020	2
101K003	5
101K011	6
101K012	2
101K013	4
101K015	1
101L003	5
101L006	1
101L008	1
101M007	5
101M010	1
101M011	1
101M017	2
101M018	8
101M019	1
101M020	2
101N003	4
101N004	3
101N010	1
101P001	1
101R004	1
101R007	2
101S003	1
101S005	4
101S007	1
101S008	1
101S010	1
101S013	3
101S014	3
101S023	1
101S026	5
101S028	2
101S033	1
101S043	4
101T001	5
101T007	1
101Y002	1
102A002	3
102A003	2
102A004	1
102A007	3
102A009	3
102A010	2

102A011	1
102A012	2
102A013	1
102A014	1
102A026	1
102B001	1
102B008	2
102B009	1
102D001	1
102D008	4
102D009	1
102E009	1
102F002	4
102G001	3
102G006	6
102H001	4
102I001	2
102I005	1
102J001	2
102J006	3
102K004	1
102L003	2
102L004	4
102M011	1
102M013	1
102M014	2
102N001	2
102N002	1
102N003	3
102N006	1
102O001	2
102O004	3
102P007	1
102P008	2
102S001	1
102S004	1
102S006	1
102S008	2
102S009	6
102S010	1
102S011	5
102S015	3
102S019	1
102S020	1
102S023	1
102S024	1
102T004	2
102T005	2
102W001	1
102W002	1
102W007	1
102Y001	2
102Y003	1
102Y004	4
102Y005	1
102Y007	6
102Y008	2
103A007	7
103B001	2
103B003	1
103B007	5
103B008	1
103C003	3
103C004	1
103D002	1
103G004	1

103H001	6
103H002	3
103I002	2
103I003	1
103I004	1
103I005	1
103K002	1
103K003	3
103L001	4
103M004	3
103M007	1
103P001	5
103R003	10
103R005	1
103S005	1
103S010	2
103S011	1
103S012	2
103S016	1
103V001	3
103W002	1
103Y004	6
104A010	3
104A011	1
104A012	1
104A013	1
104A014	1
104A015	1
104A019	4
104A020	1
104A021	1
104B001	2
104C004	8
104C011	3
104C012	4
104C014	2
104C015	2
104C016	1
104E002	3
104E005	3
104F003	3
104G002	1
104H004	1
104I002	1
104J003	5
104J004	3
104K001	3
104L005	3
104M004	5
104M010	7
104N003	1
104O004	1
104O005	1
104P003	2
104P010	3
104P017	1
104R002	2
104R007	1
104S006	1
104S007	6
104S008	4
104S015	1
104S016	1
104S017	1
104T008	3
104W002	1

104Y002	2
105A009	10
105B003	1
105B004	1
105C001	2
105C003	2
105C004	7
105D008	1
105I001	1
105I002	6
105I006	3
105J003	3
105J006	2
105K004	3
105M003	1
105M006	4
105M008	1
105M016	2
105M018	1
105O001	3
105O004	1
105R001	3
105R005	4
105R007	1
105R008	6
105S004	3
105S011	1
105S014	1
105V002	1
105Y001	2
105Z001	1
106A003	3
106B001	3
106G001	1
106G002	1
106I002	7
106I004	1
106J001	3
106K002	6
106M002	2
106M004	1
106S003	2
106S004	2
106S006	1
107A005	2
107A010	1
107B001	2
107B003	2
107B005	2
107B007	1
107C003	1
107D002	1
107F001	1
107K003	1
107L001	3
107L003	1
107M002	4
107M003	2
107M009	1
107M010	6
107M011	4
107M015	2
107M016	2
107M017	2
107M018	2
107M019	2

107M020	1
107R002	2
107S001	6
107S002	3
107S012	1
107T002	5
107T004	7
108B003	2
108C002	7
108D002	4
108D005	3
108D006	2
108F001	1
108G006	3
108H001	1
108I002	4
108I003	1
108K001	1
108K002	1
108M001	2
108M002	5
108M008	3
108M009	1
108N001	1
108P002	1
108P004	1
108S004	2
108S005	3
108S006	6
108S008	1
108S009	2
108S010	1
108S011	2
108S012	2
108U001	2
108W001	8
109I001	1
109S001	1
109S002	1
110H001	1
110S001	1
110S002	1
112A002	1
112A003	1
112B002	1
112B004	1
112C003	1
112F001	1
112G003	2
112H001	1
112K006	1
112L001	1
112P006	1
112P007	1
112P008	1
112P009	1
112P010	1
112S005	1
113C001	1
113G001	1
113Z002	1
220P001	1
314J001	1

III. Memilih data

```
In [6]: # Memilih data secara spesifik
data_terpilih = data[['Art', 'Quantity', 'Bruto', 'Net', 'Cogs', 'Profit', 'brand', 'Id'
```

IV. Perhitungan diskon

$$disc = \frac{bruto - net}{bruto} \quad (1)$$

```
In [48]: data_terpilih['discount_per_baris'] = (data_terpilih['Bruto'] - data_terpilih['Net']) /
data_terpilih['margin_per_baris'] = (data_terpilih['Net'] - data_terpilih['Cogs']) / dat
```

Terdapat data dengan nilai penjualan bruto sama dengan 0 yang menyebabkan munculnya error pada kolom discount atau infinite karena melakukan pembagian terhadap 0. Seperti dapat dilihat dari `data_terpilih.describe()` di bawah ini yang menunjukkan bahwa mean dan juga min dari kolom discount adalah -inf. Oleh karena itu pada cell selanjutnya kita akan melakukan drop terhadap baris data dengan penjualan bruto 0.

```
In [49]: # Informasi dasar data_terpilih
print(data_terpilih.describe())
```

	Quantity	Bruto	Net	Cogs	Profit \
count	296,502.00	313,186.00	313,186.00	313,186.00	313,186.00
mean	9.26	3,097,810.14	1,754,975.12	1,262,372.95	492,602.18
std	46.69	8,965,266.71	4,342,467.47	3,519,098.01	1,239,447.72
min	1.00	18,800.00	2,480.00	-663,997.65	-22,692,737.43
25%	2.00	579,800.00	384,860.00	250,005.37	118,514.04
50%	3.00	1,159,600.00	747,728.00	490,588.98	240,393.22
75%	6.00	2,399,400.00	1,553,864.00	1,032,360.04	492,052.00
max	7,800.00	499,600,000.00	262,290,000.00	184,894,004.50	94,125,000.00

	discount	frekuensi_transaksi	discount_per_baris	margin_per_baris
count	313,186.00	313,186.00	313,186.00	313,186.00
mean	0.36	1.00	0.36	0.31
std	0.10	0.00	0.10	0.29
min	-0.00	1.00	-0.00	-73.93
25%	0.31	1.00	0.31	0.29
50%	0.33	1.00	0.33	0.35
75%	0.34	1.00	0.34	0.39
max	0.99	1.00	0.99	2.49

```
In [91]: data_terpilih = data_terpilih[data_terpilih['Bruto'] != 0]

# Memisahkan data dengan penjualan normal (tingkat discount di bawah 50%) dan penjualan
data_terpilih_normal = data_terpilih[data_terpilih['discount_per_baris'] < 0.5]
data_terpilih_diskon = data_terpilih[data_terpilih['discount_per_baris'] >= 0.5]
```

IV.a Informasi semua data

```
In [89]: print(data_terpilih.describe())
```

	Quantity	Bruto	Net	Cogs	Profit \
count	296,502.00	313,186.00	313,186.00	313,186.00	313,186.00
mean	9.26	3,097,810.14	1,754,975.12	1,262,372.95	492,602.18
std	46.69	8,965,266.71	4,342,467.47	3,519,098.01	1,239,447.72
min	1.00	18,800.00	2,480.00	-663,997.65	-22,692,737.43

25%	2.00	579,800.00	384,860.00	250,005.37	118,514.04
50%	3.00	1,159,600.00	747,728.00	490,588.98	240,393.22
75%	6.00	2,399,400.00	1,553,864.00	1,032,360.04	492,052.00
max	7,800.00	499,600,000.00	262,290,000.00	184,894,004.50	94,125,000.00

	discount	frekuensi_transaksi	discount_per_baris	margin_per_baris
count	313,186.00	313,186.00	313,186.00	313,186.00
mean	0.36	1.00	0.36	0.31
std	0.10	0.00	0.10	0.29
min	-0.00	1.00	-0.00	-73.93
25%	0.31	1.00	0.31	0.29
50%	0.33	1.00	0.33	0.35
75%	0.34	1.00	0.34	0.39
max	0.99	1.00	0.99	2.49

IV.b Informasi data penjualan normal

```
In [90]: print(data_terpilih_normal.describe())
```

	Quantity	Bruto	Net	Cogs	Profit \
count	257,210.00	272,988.00	272,988.00	272,988.00	272,988.00
mean	7.21	2,284,587.38	1,532,483.19	998,190.26	534,292.94
std	34.48	5,610,917.84	3,740,023.10	2,625,503.66	1,237,525.90
min	1.00	20,000.00	15,376.00	-574,414.64	-4,233,978.33
25%	1.00	549,800.00	384,860.00	243,696.45	133,451.78
50%	2.00	1,099,600.00	736,732.00	474,996.88	258,303.36
75%	5.00	2,199,200.00	1,473,464.00	927,611.16	515,502.00
max	5,235.00	499,600,000.00	262,290,000.00	184,894,004.50	94,125,000.00

	discount	frekuensi_transaksi	discount_per_baris	margin_per_baris
count	272,988.00	272,988.00	272,988.00	272,988.00
mean	0.33	1.00	0.33	0.35
std	0.03	0.00	0.03	0.07
min	-0.00	1.00	-0.00	-0.63
25%	0.30	1.00	0.30	0.32
50%	0.33	1.00	0.33	0.36
75%	0.33	1.00	0.33	0.39
max	0.50	1.00	0.50	2.49

IV.b.1 Analisa berdasar Tingkat Diskon

```
In [95]: net_berdasar_disc = data_terpilih_normal[['discount_per_baris', 'Net']].groupby(['discount_per_baris'])
with pd.option_context('display.max_rows', None):
    print(net_berdasar_disc.sort_values(by='Net', ascending=False))
```

discount_per_baris	Net
0.33	202,435,965,038.00
0.30	78,733,998,140.00
0.35	27,205,627,020.00
0.38	21,757,632,472.00
0.40	14,558,451,180.00
0.31	13,037,910,042.00
0.32	12,208,280,448.00
0.00	6,920,325,600.00
0.35	5,578,830,234.00
0.41	4,785,928,758.00
0.27	4,445,090,158.00
0.29	3,756,455,504.00
0.28	3,189,646,440.00
0.34	2,478,582,680.00
0.37	1,961,415,666.00
0.30	1,630,494,186.00

0.45	1,263,193,030.00
0.15	1,160,522,170.00
0.34	1,082,440,920.00
0.20	1,031,604,640.00
0.49	1,003,719,618.00
0.38	959,996,125.00
0.47	852,537,000.00
0.41	588,290,661.00
0.43	559,615,493.50
0.33	518,235,435.00
0.25	388,835,700.00
0.40	376,788,114.30
0.29	304,129,950.00
0.39	292,026,032.00
0.38	231,935,717.00
0.42	223,746,020.00
0.32	217,775,611.00
0.37	203,216,707.20
0.46	159,784,917.60
0.46	153,032,888.80
0.47	149,595,239.20
0.49	140,377,296.00
0.48	133,464,790.80
0.28	117,077,495.00
0.38	116,125,776.00
0.49	102,941,080.00
0.40	102,531,600.00
0.45	99,102,960.00
0.47	81,553,008.00
0.40	75,492,384.00
0.45	72,305,341.50
0.38	59,371,200.00
0.44	59,147,668.80
0.46	54,438,692.00
0.50	45,713,467.80
0.44	37,067,106.56
0.45	36,772,430.82
0.41	34,819,200.00
0.49	34,742,614.50
0.41	31,903,092.00
0.44	31,591,284.00
0.33	28,299,516.07
0.50	28,131,364.80
0.43	25,573,514.40
0.40	25,473,146.40
0.34	24,101,904.00
0.41	22,868,551.52
0.42	22,800,476.00
0.48	22,201,106.40
0.41	18,903,924.00
0.44	17,269,901.92
0.41	14,032,974.80
0.37	13,305,567.60
0.40	12,848,640.00
0.41	11,954,015.56
0.41	11,011,572.00
0.42	10,935,859.20
0.44	10,895,623.20
0.49	10,808,826.00
0.36	10,764,672.00
0.41	9,575,193.60
0.41	8,835,576.72
0.41	8,704,800.00
0.33	8,399,856.36
0.46	7,005,026.60
0.47	6,847,669.40

0.39	6,510,336.00
0.39	5,469,621.36
0.41	5,197,398.07
0.33	5,000,000.00
0.41	4,937,528.17
0.37	4,836,187.20
0.49	4,632,348.60
0.50	4,537,533.00
0.42	4,512,436.80
0.50	4,250,000.00
0.41	3,898,048.55
0.40	3,287,029.20
0.41	3,191,731.20
0.39	3,018,876.00
0.50	2,829,927.10
0.43	2,805,000.00
0.48	2,481,149.52
0.48	2,481,149.52
0.43	2,189,527.20
0.41	2,109,294.00
0.48	1,455,695.34
0.23	1,400,000.00
0.43	1,085,918.40
0.50	800,000.00
-0.00	700,000.00
0.50	600,000.00
0.48	600,000.00
0.48	327,372.76
0.40	300,000.00
0.50	194,894.70
0.30	140,000.00

```
In [96]: print(net_berdasar_disc.describe())
```

	Net
count	113.00
mean	3,702,208,162.16
std	20,603,248,895.28
min	140,000.00
25%	5,197,398.07
50%	31,591,284.00
75%	304,129,950.00
max	202,435,965,038.00

IV.b.2 Analisa berdasar Customer

```
In [97]: net_berdasar_cust = data_terpilih_normal[['Id Cust', 'Net']].groupby(['Id Cust']).sum()
```

IV.b.2.1 Total nilai Net Sales

```
In [98]: net_berdasar_cust.sum()
```

```
Out[98]: Net    418,349,522,324.62
dtype: float64
```

IV.b.2.2 Informasi Net Sales berdasar pengelompokan Customer

```
In [99]: print(net_berdasar_cust.describe())
```

	Net
count	344.00
mean	1,216,132,332.34
std	3,123,164,811.92

min	39,800.00
25%	43,674,330.00
50%	226,606,467.00
75%	1,031,640,159.00
max	36,220,967,218.00

IV.b.2.3 Customer dengan nilai total penjualan di atas rata - rata nilai penjualan

Berikut adalah kontribusi net sales 2020 - 2022 YTD untuk customer dengan total nilai net sales di atas nilai rata - rata net sales dari semua customer (kecuali PT Prestasi Retail Innovation - 220P001)

```
In [100... # Menslice customer dengan total nilai Net Sales di atas rata - rata Net Sales
cust_diatas_rerata = net_berdasar_cust.loc[(net_berdasar_cust['Net'] > net_berdasar_cust
                                         (net_berdasar_cust.index != "220P001"))].sort_
net_mean = net_berdasar_cust['Net'].mean()
total_cust = net_berdasar_cust['Net'].count()
total_cust_diatas_rerata = cust_diatas_rerata['Net'].count()

with pd.option_context('display.max_rows', None):
    print(f'Berikut adalah daftar customer dengan total penjualan di atas rata - rata to
          f'Jumlah customer di atas rata - rata {total_cust_diatas_rerata} orang dari to
```

Berikut adalah daftar customer dengan total penjualan di atas rata - rata total penjuala
n periode 2020 - 2022 YTD (Rp 1,216,132,332):

Jumlah customer di atas rata - rata 77 orang dari total customer 344 orang

Net

Id Cust

104A019	36,220,967,218.00
101G012	23,398,958,242.00
105A009	23,021,827,330.00
101I020	10,968,703,171.00
101M018	10,239,439,158.00
101K003	9,654,975,979.00
102L004	9,243,403,779.50
102S008	8,668,741,114.00
101K011	8,063,359,046.00
101D010	7,332,266,088.00
104C004	7,293,818,258.00
101F003	6,990,350,258.00
103P001	6,751,407,288.00
107M002	6,606,956,623.00
102S009	6,573,863,592.00
102G006	6,268,568,294.00
102M014	6,220,965,438.00
102I005	6,098,193,000.00
102S015	5,888,689,001.16
102P008	5,578,720,454.00
101S014	5,108,099,620.80
101S005	4,871,792,924.00
102A014	4,438,285,694.00
102A004	4,346,395,686.00
104S016	4,244,198,094.00
105O004	4,225,694,512.00
101M017	3,854,485,098.00
112A002	3,853,938,423.43
102I001	3,636,460,522.00
101K013	3,564,829,201.00
101T001	3,355,124,812.00
107M018	3,228,989,949.00
105J006	3,109,636,642.00
112A003	3,047,100,000.00
102O004	3,029,368,480.00
108W001	2,970,737,516.00
104J003	2,950,701,176.00
104T008	2,878,085,000.00

101N003	2,876,555,830.80
101D003	2,840,658,115.00
112F001	2,490,105,903.39
101L003	2,403,040,708.00
101F015	2,353,120,474.00
102Y004	2,343,934,220.00
101M020	2,190,342,136.00
104S007	2,098,442,707.00
102A026	2,084,227,613.00
106K002	2,081,382,532.00
102M013	2,022,475,809.00
103B003	1,924,554,508.00
108S008	1,920,254,318.00
101M007	1,913,297,076.00
102T004	1,876,957,932.00
102Y007	1,821,430,686.00
107S001	1,791,609,954.00
104A010	1,775,954,540.00
102S023	1,769,823,294.00
101B022	1,742,329,638.00
101A005	1,704,407,104.00
104R007	1,669,099,054.00
103A007	1,636,820,522.00
101N004	1,631,472,410.00
102D001	1,631,214,656.00
104L005	1,585,498,988.00
107M011	1,566,581,628.00
108I002	1,558,655,582.40
103R003	1,543,473,848.00
101G018	1,522,878,416.00
102N003	1,437,427,181.00
101S043	1,415,660,668.00
101B003	1,361,976,888.00
108S006	1,323,699,889.10
103S012	1,279,607,782.00
103H002	1,272,347,021.00
101B013	1,245,462,426.00
101B024	1,243,759,364.00
108S012	1,227,172,838.00

In [101...

```
plt.figure(figsize=(15,10))
plt.barh(cust_diatas_rerata.index, cust_diatas_rerata['Net'], color='purple')
plt.xlabel("Net Sales", fontsize=14)
plt.title(f"Net Sales Customer dengan Total Nilai Net Sales di atas Rata - Rata (Rp {for
plt.gca().xaxis.set_major_formatter(ticker.StrMethodFormatter('{x:,.0f}'))
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['bottom'].set_visible(False)
plt.grid(False)
plt.yticks(fontsize=5)
plt.xticks(fontsize=9)
plt.show()
```

Customer ID	Net Sales (Rp)
108B012	1,500,000,000
101B024	1,500,000,000
101B013	1,500,000,000
103B002	1,500,000,000
103B012	1,500,000,000
108S006	1,500,000,000
101B003	1,500,000,000
101S043	1,500,000,000
102N003	1,500,000,000
101G018	1,500,000,000
103R003	1,500,000,000
108B002	1,500,000,000
107M011	1,500,000,000
104L005	1,500,000,000
102D001	1,500,000,000
101N004	1,500,000,000
103A007	1,500,000,000
104B007	1,500,000,000
101A006	1,500,000,000
101B022	1,500,000,000
102S023	1,500,000,000
104A010	1,500,000,000
107S001	1,500,000,000
102Y007	1,500,000,000
102T004	1,500,000,000
101M007	1,500,000,000
108S008	1,500,000,000
103B003	1,500,000,000
102M013	1,500,000,000
104M002	1,500,000,000
102A026	1,500,000,000
104S007	1,500,000,000
101M020	1,500,000,000
102Y004	1,500,000,000
101P015	1,500,000,000
101L003	1,500,000,000
112P001	1,500,000,000
101D003	1,500,000,000
101N003	1,500,000,000
104T008	1,500,000,000
104A003	1,500,000,000
108W001	1,500,000,000
102C004	1,500,000,000
112A003	1,500,000,000
105J006	1,500,000,000
107M018	1,500,000,000
101T001	1,500,000,000
101K013	1,500,000,000
102B001	1,500,000,000
112A002	1,500,000,000
101M017	1,500,000,000
102C004	1,500,000,000
104S016	1,500,000,000
102A004	1,500,000,000
102A014	1,500,000,000
101S005	1,500,000,000
101S014	1,500,000,000
102P008	1,500,000,000
102S015	1,500,000,000
102D005	1,500,000,000
102M014	1,500,000,000
102C006	1,500,000,000
102S009	1,500,000,000
107M002	1,500,000,000
103P001	1,500,000,000
101P003	1,500,000,000
104C004	1,500,000,000
101D010	1,500,000,000
101K011	1,500,000,000
102B008	1,500,000,000
108L004	1,500,000,000
101K003	1,500,000,000
101M018	1,500,000,000
101B020	1,500,000,000
108A009	1,500,000,000
101G012	1,500,000,000
104A019	35,000,000,000

IV.b.2.4 Pengelompokan dan penjumlahan Net Sales berdasarkan Top 2 Customer dan Tingkat Diskon

In [102...

```
net_berdasar_disc_cust = data_terpilih_normal[['Id Cust', 'discount_per_baris', 'Net']].
print(net_berdasar_disc_cust.iloc[net_berdasar_disc_cust.index.get_level_values('Id Cust')])
print(net_berdasar_disc_cust.iloc[net_berdasar_disc_cust.index.get_level_values('Id Cust')])
```

		Net
Id Cust	discount_per_baris	
104A019	0.30	11,294,376,380.00
	0.32	88,334,992.00
	0.33	18,529,096,024.00
	0.35	2,427,835,410.00
	0.38	1,764,045,824.00
	0.40	611,094,480.00
	0.41	244,246,860.00
	0.45	258,217,630.00
	0.49	1,003,719,618.00

		Net
Id Cust	discount_per_baris	
101G012	0.15	245,636,400.00
	0.30	5,014,764,580.00
	0.33	14,625,286,352.00
	0.34	8,522,712.00
	0.35	1,313,725,010.00
	0.38	71,462,016.00
	0.38	1,317,023,096.00
	0.38	13,278,720.00
	0.40	309,808,800.00
	0.41	479,450,556.00

IV.b.3 Analisa Berdasar Tingkat Diskon

```
In [103]: disc_rerata_cust = data_terpilih_normal[['Id Cust', 'Quantity', 'Bruto', 'Net', 'Cogs',
disc_rerata_cust['rerata_disc'] = (disc_rerata_cust['Bruto'] - disc_rerata_cust['Net'])
disc_rerata_cust['rerata_margin'] = (disc_rerata_cust['Net'] - disc_rerata_cust['Cogs'])
disc_rerata_cust['rerata_spending'] = (disc_rerata_cust['Bruto'] / disc_rerata_cust['Qua
disc_rerata_cust['ATV'] = disc_rerata_cust['Net'] / disc_rerata_cust['frekuensi_transaks
disc_rerata_cust.sort_values(by='rerata_disc', ascending=False)
disc_rerata_cust
```

Out[103]:

	Quantity	Bruto	Net	Cogs	frekuensi_transaksi	rerata_disc	rerata_mar
Id Cust							
1.01E+07	707.00	259,268,800.00	183,149,668.00	115,222,318.19	336	0.29	0
1.01E+08	30.00	35,994,000.00	28,795,200.00	8,671,489.81	1	0.20	0
1.02E+11	3,234.00	815,694,800.00	543,709,425.00	345,330,522.65	586	0.33	0
1.04E+04	286.00	102,802,800.00	67,507,106.00	40,759,062.94	83	0.34	0
1.04E+07	1,664.00	642,443,200.00	426,961,036.00	299,048,119.50	388	0.34	0
...
112S005	20.00	35,996,000.00	25,197,200.00	12,391,496.68	6	0.30	0
113C001	170.00	52,686,000.00	48,929,370.00	14,071,364.66	65	0.07	0
113G001	184.00	36,833,200.00	25,783,240.00	14,171,649.28	29	0.30	0
113Z002	1.00	39,800.00	39,800.00	14,774.58	1	0.00	0
220P001	6,080.00	7,523,504,000.00	4,859,939,941.24	4,646,870,657.49	663	0.35	0

344 rows × 9 columns

```
In [104]: disc_rerata_cust.nunique()
```

Out[104]:

Quantity	322
Bruto	342
Net	342
Cogs	344
frekuensi_transaksi	276
rerata_disc	286
rerata_margin	344
rerata_spending	336
ATV	343
dtype:	int64

```
In [105]: disc_rerata_cust['rerata_disc'].unique()
```

Out[105]:

```
array([0.29359156, 0.2          , 0.33344012, 0.34333398, 0.33541045,
        0.29618394, 0.33093484, 0.32983794, 0.27          , 0.315          ,
        0.34608262, 0.34          , 0.32786861, 0.32856215, 0.32628891,
        0.31356038, 0.33433225, 0.30169461, 0.28996271, 0.30480576,
        0.31486394, 0.27678458, 0.29          , 0.27992587, 0.34219659,
        0.3998758 , 0.31869555, 0.28364003, 0.3          , 0.31738209,
        0.32462857, 0.25269871, 0.32851611, 0.28          , 0.32978424,
        0.31621951, 0.37319185, 0.34237269, 0.31277762, 0.4          ,
        0.27360155, 0.32          , 0.33688575, 0.32740522, 0.33822107,
        0.30239376, 0.31172323, 0.30596208, 0.30716244, 0.32463824,
        0.31710633, 0.31          , 0.3222571 , 0.32721458, 0.30456667,
        0.38340906, 0.32842999, 0.33017032, 0.32993148, 0.30259363,
        0.34688197, 0.31957859, 0.2941465 , 0.33730822, 0.29472706,
        0.31563849, 0.32187899, 0.32347048, 0.3331015 , 0.27390617,
        0.31090183, 0.3237695 , 0.33          , 0.35          , 0.34639586,
        0.31097227, 0.33756803, 0.2909088 , 0.33362298, 0.34966936,
```



```

0.33266416, 0.32758184, 0.32242477, 0.34912249, 0.31463954,
0.33163109, 0.32232944, 0.33285373, 0.31342045, 0.31798268,
0.3535672 , 0.31310226, 0.32813209, 0.33665623, 0.33442116,
0.33574387, 0.34952525, 0.33544203, 0.30789979, 0.32846052,
0.34069955, 0.31438179, 0.3364658 , 0.31020389, 0.32154197,
0.30911261, 0.32846078, 0.27428898, 0.35328607, 0.25 ,
0.32338271, 0.33299098, 0.33128941, 0.34008885, 0.33274008,
0.30445374, 0.34900376, 0.34609433, 0.33177027, 0.33501368,
0.32721824, 0.2996597 , 0.28225451, 0.33651817, 0.34728609,
0.33817418, 0.31705603, 0.33047547, 0.33359803, 0.32765307,
0.32481867, 0.32869597, 0.30448443, 0.33103309, 0.33089467,
0.32287887, 0.32268495, 0.32845191, 0.32925677, 0.32997564,
0.27969791, 0.30584471, 0.27008754, 0.32246885, 0.2965402 ,
0.27341526, 0.3320588 , 0.34109378, 0.33533635, 0.32033751,
0.3298624 , 0.32221658, 0.32992623, 0.39006579, 0.32819383,
0.33409406, 0.33270984, 0.28769122, 0.3394261 , 0.34150329,
0.31983613, 0.3039241 , 0.33237066, 0.32180842, 0.33273894,
0.34115927, 0.34609969, 0.41665739, 0.33824498, 0.28793046,
0.32149409, 0.32117191, 0.31945594, 0.33265969, 0.31861069,
0.32136707, 0.3121405 , 0.35965571, 0.29752425, 0.29612192,
0.3379264 , 0.35273129, 0.30766675, 0.40468216, 0.33709378,
0.33081884, 0.31935643, 0.32986463, 0.31008635, 0.31550937,
0.29419993, 0.33178029, 0.28739748, 0.31312398, 0.3158386 ,
0.30208351, 0.34311962, 0.31899198, 0.31017864, 0.32123423,
0.28796927, 0.33255134, 0.34462578, 0.29465162, 0.28997019,
0.30649878, 0.31694758, 0.31103273, 0.28670733, 0.34114022,
0.28952315, 0.35027047, 0.32362275, 0.33329918, 0.31238701,
0.30227348, 0.30082065, 0.30585551, 0.26514159, 0.33006182,
0.33114983, 0.38462073, 0.32083105, 0.32021035, 0.34122108,
0.33256103, 0.34205668, 0.35474887, 0.33097378, 0.30625987,
0.33638266, 0.31190884, 0.33170241, 0.31127434, 0.3431257 ,
0.33746628, 0.33776222, 0.33782948, 0.32972961, 0.33545526,
0.33936984, 0.27566825, 0.32728435, 0.32069654, 0.30839688,
0.33590102, 0.34133332, 0.35837488, 0.34328167, 0.32776412,
0.33819164, 0.33151756, 0.36266259, 0.3410686 , 0.34006182,
0.33868803, 0.31611002, 0.415 , 0.34280392, 0.32621253,
0.33421104, 0.34686947, 0.32790631, 0.3390701 , 0.34102916,
0.32940016, 0.33728631, 0.30333587, 0.33308139, 0.33949373,
0.33811352, 0.33399973, 0.34378125, 0.30604095, 0.33957267,
0.30312188, 0.00369247, 0. , 0.34131728, 0.34307938,
0.32174245, 0.26227334, 0.23683268, 0.3416684 , 0.07130224,
0.35403238])

```

```

In [106.. fig = plt.figure(figsize=(15, 4))

ax1 = fig.add_subplot(131)
ax2 = fig.add_subplot(132)
ax3 = fig.add_subplot(133)

ax1.scatter(disc_rerata_cust['rerata_disc'], disc_rerata_cust['rerata_spending'], c='purple')
ax1.yaxis.set_major_formatter(ticker.StrMethodFormatter('{x:,.0f}'))
ax1.set_title('Distribusi Customer Berdasar Diskon VS Spending', fontsize=10)
ax1.set_xlabel('Rerata Diskon', fontsize=7)
ax1.set_ylabel('Rerata Spending', fontsize=7)
ax1.xaxis.set_tick_params(labelsize=7)
ax1.yaxis.set_tick_params(labelsize=7)

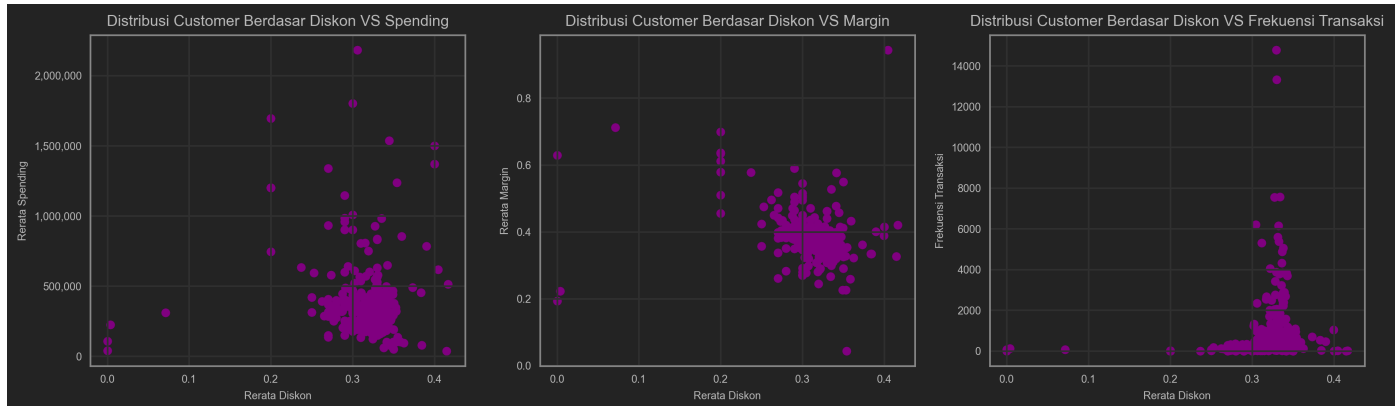
ax2.scatter(disc_rerata_cust['rerata_disc'], disc_rerata_cust['rerata_margin'], c='purple')
# ax2.yaxis.set_major_formatter(ticker.StrMethodFormatter('{x:,.0f}'))
ax2.set_title('Distribusi Customer Berdasar Diskon VS Margin', fontsize=10)
ax2.set_xlabel('Rerata Diskon', fontsize=7)
ax2.set_ylabel('Rerata Margin', fontsize=7)
ax2.xaxis.set_tick_params(labelsize=7)
ax2.yaxis.set_tick_params(labelsize=7)

```

```

ax3.scatter(disc_rerata_cust['rerata_disc'], disc_rerata_cust['frekuensi_transaksi'], c=
# ax3.yaxis.set_major_formatter(ticker.StrMethodFormatter('{x:,.0f}'))
ax3.set_title('Distribusi Customer Berdasar Diskon VS Frekuensi Transaksi', fontsize=10)
ax3.set_xlabel('Rerata Diskon', fontsize=7)
ax3.set_ylabel('Frekuensi Transaksi', fontsize=7)
ax3.xaxis.set_tick_params(labelsize=7)
ax3.yaxis.set_tick_params(labelsize=7)
plt.show()

```



```

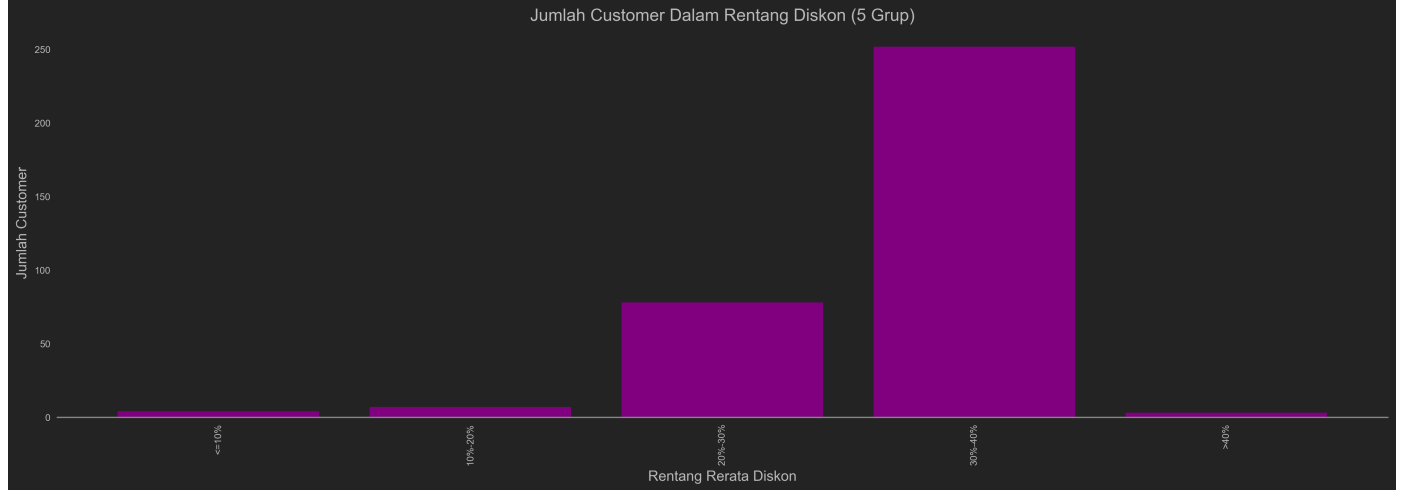
In [111... # pengelompokan berdasar range diskon
n_grup = 5
batas_atas = 0.5 # berdasarkan tingkat diskon penjualan normal

x = []
y = []

for numerator in range(n_grup):
    if numerator == 0:
        x.append("<=" + format(((numerator + 1) / n_grup) * batas_atas, '.0%'))
        y.append(len(disc_rerata_cust[disc_rerata_cust['rerata_disc'] <= ((numerator + 1)
    elif numerator + 1 == n_grup:
        x.append(">" + format(((numerator) / n_grup) * batas_atas, '.0%'))
        y.append(len(disc_rerata_cust[disc_rerata_cust['rerata_disc'] > ((numerator) / n
    else:
        x.append(format(((numerator) / n_grup) * batas_atas, '.0%') + "-" + format(((num
        y.append(len(disc_rerata_cust[(disc_rerata_cust['rerata_disc'] > (numerator / n_

plt.figure(figsize=(20, 6))
plt.bar(x,y, color='purple')
plt.grid(False)
plt.gca().spines['left'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.title(f'Jumlah Customer Dalam Rentang Diskon ({n_grup} Grup)')
plt.xlabel("Rentang Rerata Diskon", fontsize=12)
plt.ylabel("Jumlah Customer", fontsize=12)
plt.xticks(fontsize=8, rotation='vertical')
plt.yticks(fontsize=8)
plt.show()

```



IV.c Informasi data penjualan diskon

```
In [93]: print(data_terpilih_diskon.describe())
```

	Quantity	Bruto	Net	Cogs	Profit \
count	39,292.00	40,198.00	40,198.00	40,198.00	40,198.00
mean	22.65	8,620,474.30	3,265,936.54	3,056,459.80	209,476.74
std	91.99	19,427,655.85	7,021,854.55	6,780,907.16	1,215,167.12
min	1.00	18,800.00	2,480.00	-663,997.65	-22,692,737.43
25%	2.00	999,800.00	399,200.00	363,762.71	-20,992.57
50%	5.00	2,499,000.00	963,336.00	885,980.04	59,391.00
75%	16.00	7,498,000.00	2,998,500.00	2,723,261.58	247,366.10
max	7,800.00	469,747,800.00	153,753,600.00	146,958,325.09	32,250,350.87

	discount	frekuensi_transaksi	discount_per_baris	margin_per_baris
count	40,198.00	40,198.00	40,198.00	40,198.00
mean	0.61	1.00	0.61	0.01
std	0.09	0.00	0.09	0.72
min	0.50	1.00	0.50	-73.93
25%	0.53	1.00	0.53	-0.03
50%	0.56	1.00	0.56	0.09
75%	0.70	1.00	0.70	0.18
max	0.99	1.00	0.99	1.51