In [7]:
```
#import library yang dibutuhkan
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
Matplotlib is building the font cache; this may take a moment.
```

GATHERING DATA

In [8]:
```
#data customers
customers_df = pd.read_csv("https://raw.githubusercontent.com/dicodingacademy/dicod
customers_df.head()
```

Out[8]:

	customer_id	customer_name	gender	age	home_address	zip_code	city	
0	1	fulan 1	Female	30	8606 Victoria TerraceSuite 560	5464	Johnstonhaven	I
1	2	fulan 2	Prefer not to say	69	8327 Kirlin SummitApt. 461	8223	New Zacharyfort	
2	3	fulan 3	Prefer not to say	59	269 Gemma SummitSuite 109	5661	Aliburgh	A
3	4	fulan 4	Prefer not to say	67	743 Bailey GroveSuite 141	1729	South Justinhaven	Qu
4	5	fulan 5	Prefer not to say	30	48 Hyatt ManorSuite 375	4032	Griffithsshire	Qu

In [9]:
```
#load tabel orders
orders_df = pd.read_csv("https://raw.githubusercontent.com/dicodingacademy/dicoding
orders_df.head()
```

Out[9]:

	order_id	customer_id	payment	order_date	delivery_date
0	1	64	30811	2021-8-30	2021-09-24
1	2	473	50490	2021-2-3	2021-02-13
2	3	774	46763	2021-10-8	2021-11-03
3	4	433	39782	2021-5-6	2021-05-19
4	5	441	14719	2021-3-23	2021-03-24

In [10]:
```
#tabel product
product_df = pd.read_csv("https://raw.githubusercontent.com/dicodingacademy/dicodin
product_df.head()
```

Out[10]:

	product_id	product_type	product_name	size	colour	price	quantity	description
0	0	Shirt	Oxford Cloth	XS	red	114	66	A red coloured, XS sized, Oxford Cloth Shirt
1	1	Shirt	Oxford Cloth	S	red	114	53	A red coloured, S sized, Oxford Cloth Shirt
2	2	Shirt	Oxford Cloth	M	red	114	54	A red coloured, M sized, Oxford Cloth Shirt
3	3	Shirt	Oxford Cloth	L	red	114	69	A red coloured, L sized, Oxford Cloth Shirt
4	4	Shirt	Oxford Cloth	XL	red	114	47	A red coloured, XL sized, Oxford Cloth Shirt

In [11]:
```
#tabel sales
sales_df = pd.read_csv("https://raw.githubusercontent.com/dicodingacademy/dicoding_
sales_df.head()
```

Out[11]:

	sales_id	order_id	product_id	price_per_unit	quantity	total_price
0	0	1	218	106	2	212.0
1	1	1	481	118	1	118.0
2	2	1	2	96	3	288.0
3	3	1	1002	106	2	212.0
4	4	1	691	113	3	339.0

ASSESING DATA

In [12]:
```
#Menilai/Memeriksa Data customers_df
customers_df.info()
'''dari hasil diperoleh ada missing value pada kolom gender'''
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1007 entries, 0 to 1006
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   customer_id    1007 non-null   int64 
 1   customer_name  1007 non-null   object
 2   gender         989 non-null    object
 3   age            1007 non-null   int64 
 4   home_address   1007 non-null   object
 5   zip_code       1007 non-null   int64 
 6   city           1007 non-null   object
 7   state          1007 non-null   object
 8   country        1007 non-null   object
dtypes: int64(3), object(6)
memory usage: 70.9+ KB
```

In [13]:
```
customers_df.isna().sum()
'''ada 18 missing value'''
```

Out[13]:
```
customer_id       0
customer_name     0
gender           18
age               0
home_address      0
zip_code          0
city              0
state             0
country           0
dtype: int64
```

In [14]:
```
print("Jumlah duplikasi: ", customers_df.duplicated().sum())
```

```
Jumlah duplikasi:  6
```

In [15]:
```
customers_df.describe()
'''ada inaccurate value pada kolom age'''
```

Out[15]:

	customer_id	age	zip_code
count	1007.000000	1007.000000	1007.000000
mean	501.726912	50.929494	5012.538232
std	288.673238	30.516299	2885.836112
min	1.000000	20.000000	2.000000
25%	252.500000	34.000000	2403.500000
50%	502.000000	50.000000	5087.000000
75%	751.500000	65.000000	7493.500000
max	1000.000000	700.000000	9998.000000

In [16]:
```
#Menilai Data orders_df
orders_df.info()
'''ada kesalahan tipe data pada kolom order data & delivery data'''
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   order_id       1000 non-null   int64 
 1   customer_id    1000 non-null   int64 
 2   payment        1000 non-null   int64 
 3   order_date     1000 non-null   object
 4   delivery_date  1000 non-null   object
dtypes: int64(3), object(2)
memory usage: 39.2+ KB
```

In [17]:
```
print("Jumlah duplikasi: ", orders_df.duplicated().sum())
orders_df.describe()
```

```
Jumlah duplikasi:  0
```

Out[17]:
```
<bound method NDFrame.describe of      order_id  customer_id  payment order_date d
elivery_date
0           1           64    30811  2021-8-30    2021-09-24
1           2          473    50490   2021-2-3    2021-02-13
2           3          774    46763  2021-10-8    2021-11-03
3           4          433    39782   2021-5-6    2021-05-19
4           5          441    14719  2021-3-23    2021-03-24
..        ...          ...      ...        ...           ...
995       996          345    37843  2021-1-13    2021-02-02
996       997          346    53831  2021-1-18    2021-01-31
997       998          407    53308   2021-5-5    2021-05-21
998       999          428    31643  2021-6-15    2021-07-12
999      1000          896    27836   2021-4-7    2021-04-24

[1000 rows x 5 columns]>
```

In [18]:
```
#Menilai Data Product
product_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1266 entries, 0 to 1265
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   product_id    1266 non-null   int64 
 1   product_type  1266 non-null   object
 2   product_name  1266 non-null   object
 3   size          1266 non-null   object
 4   colour        1266 non-null   object
 5   price         1266 non-null   int64 
 6   quantity      1266 non-null   int64 
 7   description   1266 non-null   object
dtypes: int64(3), object(5)
memory usage: 79.3+ KB
```

In [19]:
```
print("Jumlah duplikasi: ", product_df.duplicated().sum())
product_df.describe()
'''ada 6 duplikat'''
```

```
Jumlah duplikasi:  6
```

Out[19]:

	product_id	price	quantity
count	1266.000000	1266.000000	1266.000000
mean	627.926540	105.812006	60.138231
std	363.971586	9.715611	11.682791
min	0.000000	90.000000	40.000000
25%	313.250000	95.250000	50.000000
50%	626.500000	109.000000	60.000000
75%	942.750000	114.000000	70.000000
max	1259.000000	119.000000	80.000000

In [20]:
```
#Menilai data sales_df
sales_df.info()
'''ada missing value pada kolom total_price'''
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   sales_id        5000 non-null   int64  
 1   order_id        5000 non-null   int64  
 2   product_id      5000 non-null   int64  
 3   price_per_unit  5000 non-null   int64  
 4   quantity        5000 non-null   int64  
 5   total_price     4981 non-null   float64
dtypes: float64(1), int64(5)
memory usage: 234.5 KB
```

In [21]:
```
sales_df.isna().sum()
'''ada 19 missing value pada kolom total_price'''
```

Out[21]:
```
sales_id           0
order_id           0
product_id         0
price_per_unit     0
quantity           0
total_price       19
dtype: int64
```

In [22]:
```
print("Jumlah duplikasi: ", sales_df.duplicated().sum())
sales_df.describe()
```

```
Jumlah duplikasi:  0
```

Out[22]:

	sales_id	order_id	product_id	price_per_unit	quantity	total_price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.00000	4981.000000
mean	2499.500000	503.038200	634.053200	103.501600	1.99240	206.307368
std	1443.520003	285.964418	363.255794	9.195004	0.80751	86.352449
min	0.000000	1.000000	1.000000	90.000000	1.00000	90.000000
25%	1249.750000	258.000000	323.000000	95.000000	1.00000	112.000000
50%	2499.500000	504.500000	635.000000	102.000000	2.00000	204.000000
75%	3749.250000	749.000000	951.000000	112.000000	3.00000	285.000000
max	4999.000000	999.000000	1259.000000	119.000000	3.00000	357.000000

DATA CLEANING

In [23]:
```
#Data customers_df : duplicate data, missing value, dan inaccurate value.

#menghilangkah duplikat data
customers_df.drop_duplicates(inplace=True)
print("Jumlah duplikasi : ", customers_df.duplicated().sum())
```

```
Jumlah duplikasi :  0
```

In [24]:
```
#Missing Valur customers_df
customers_df[customers_df.gender.isna()]
```

Out[24]:

	customer_id	customer_name	gender	age	home_address	zip_code	city
38	39	fulan 39	NaN	80	7440 Cameron Estate DrSuite 628	4622	North Victoriachester
167	168	fulan 168	NaN	27	2781 Berge MallSuite 452	1975	North Leoburgh
322	322	fulan 322	NaN	30	593 Becker CircleApt. 333	1640	Jacobiview
393	393	fulan 393	NaN	34	5158 Levi HillSuite 531	1474	Johnsburgh
442	442	fulan 442	NaN	26	5157 Feil RoadApt. 633	7249	Port Chloe
722	720	fulan 720	NaN	40	31 Jordan ParadeApt. 400	1380	West Henry
745	743	fulan 743	NaN	57	09 Christopher StreetSuite 967	6226	Lake Lukemouth
773	771	fulan 771	NaN	74	7367 Wright JunctionApt. 773	8882	Kuhntown
798	795	fulan 795	NaN	49	487 Summer MewsApt. 874	1712	East Hayden
801	798	fulan 798	NaN	56	27 Aiden KnollApt. 875	6531	Port Sam
825	822	fulan 822	NaN	59	41 Jenkins KnollSuite 438	2588	Lake Andrewport
859	855	fulan 855	NaN	55	603 O'keefe KnollSuite 782	8822	Port Dylanmouth
863	859	fulan 859	NaN	38	32 Isla GroveApt. 078	7711	Rosechester
914	909	fulan 909	NaN	62	976 Murray Station StApt. 036	3227	Langfort
934	929	fulan 929	NaN	68	394 Lily HillSuite 153	2353	Beahanfurt
948	943	fulan 943	NaN	64	3117 Heller PlaceSuite 149	822	North Elijah
952	946	fulan 946	NaN	24	8227 Nicholas HillSuite 150	115	South Jasper

	customer_id	customer_name	gender	age	home_address	zip_code	city
994	988	fulan 988	NaN	35	1130 Turner Estate DrSuite 925	9386	New Harry

In [25]:
```
#baris data tersebut masih mengandung banyak informasi penting sehingga sayang jika
customers_df.gender.value_counts()
```

Out[25]:
```
gender
Prefer not to say    725
Male                 143
Female               115
Name: count, dtype: int64
```

In [27]:
```
customers_df.fillna(value="Prefer not to say", inplace=True)
customers_df.isna().sum()
```

Out[27]:
```
customer_id      0
customer_name    0
gender           0
age              0
home_address     0
zip_code         0
city             0
state            0
country          0
dtype: int64
```

In [28]:
```
#inaccurate value kolom age
customers_df[customers_df.age==customers_df.age.max()]
```

Out[28]:

	customer_id	customer_name	gender	age	home_address	zip_code	city	state
967	961	fulan 961	Prefer not to say	700	29 Farrell ParadeSuite 818	6528	New Joseph	South Australia

In [35]:
```
#terjadi human error kelebihan memasukan angka 0
customers_df.replace({'age': {customers_df['age'].max(): 50}}, inplace=True)
customers_df[customers_df.age == customers_df.age.max()]
```

Out[35]:

	customer_id	customer_name	gender	age	home_address	zip_code	city
7	8	fulan 8	Prefer not to say	75	383 Muller SummitSuite 809	7681	Samside
15	16	fulan 16	Male	75	424 Mason PlaceApt. 181	6438	New Kai
24	25	fulan 25	Prefer not to say	75	02 Gabriella PlazaApt. 474	9311	Olivershire
36	37	fulan 37	Male	75	3307 Walsh JunctionSuite 233	9751	Violetville
63	64	fulan 64	Prefer not to say	75	4927 Alice MeadowApt. 960	7787	Sanfordborough
74	75	fulan 75	Prefer not to say	75	96 Caitlin HillSuite 366	9777	Smythland
98	99	fulan 99	Prefer not to say	75	468 Shields CircleApt. 480	296	Port Madelineberg
116	117	fulan 117	Female	75	0544 Zoe CourtSuite 153	2398	New Ellieland
141	142	fulan 142	Prefer not to say	75	607 Parisian AvenueSuite 494	6043	Chelseaview
224	225	fulan 225	Prefer not to say	75	651 Garden CourtApt. 769	4755	Muellerfurt
302	303	fulan 303	Prefer not to say	75	334 Olivia MeadowSuite 855	8400	North Paige
332	332	fulan 332	Male	75	011 Hamilton PlaceSuite 215	5702	East George
338	338	fulan 338	Female	75	899 Lynch LaneSuite 349	2652	Monahanberg
391	391	fulan 391	Male	75	4592 Isabella BoulevardApt. 793	7524	North Sophieland
409	409	fulan 409	Male	75	553 Alexander TrailSuite 694	134	New Chase

	customer_id	customer_name	gender	age	home_address	zip_code	city
439	439	fulan 439	Female	75	378 John MallSuite 835	3142	West Thomas
512	512	fulan 512	Prefer not to say	75	158 Joseph LaneApt. 252	3394	Clarkemouth
540	540	fulan 540	Prefer not to say	75	297 Ferry LaneApt. 061	8550	East Amelia
579	578	fulan 578	Prefer not to say	75	256 Andrew CrestApt. 193	8281	East Marcus
605	604	fulan 604	Prefer not to say	75	17 Connor ParadeApt. 442	4324	Conroymouth
629	628	fulan 628	Prefer not to say	75	774 Christiansen StreetSuite 261	2197	West Sophie
646	645	fulan 645	Female	75	988 Jacob CircuitSuite 385	1940	Greenfeldershire
963	957	fulan 957	Male	75	5215 Mitchell TrackSuite 778	4220	Samanthaberg
971	965	fulan 965	Prefer not to say	75	4627 Paige KnollApt. 635	5542	New Callumtown

In [36]:
```
customers_df[customers_df.age == customers_df.age.max()]
```

Out[36]:

	customer_id	customer_name	gender	age	home_address	zip_code	city
7	8	fulan 8	Prefer not to say	75	383 Muller SummitSuite 809	7681	Samside
15	16	fulan 16	Male	75	424 Mason PlaceApt. 181	6438	New Kai
24	25	fulan 25	Prefer not to say	75	02 Gabriella PlazaApt. 474	9311	Olivershire
36	37	fulan 37	Male	75	3307 Walsh JunctionSuite 233	9751	Violetville
63	64	fulan 64	Prefer not to say	75	4927 Alice MeadowApt. 960	7787	Sanfordborough
74	75	fulan 75	Prefer not to say	75	96 Caitlin HillSuite 366	9777	Smythland
98	99	fulan 99	Prefer not to say	75	468 Shields CircleApt. 480	296	Port Madelineberg
116	117	fulan 117	Female	75	0544 Zoe CourtSuite 153	2398	New Ellieland
141	142	fulan 142	Prefer not to say	75	607 Parisian AvenueSuite 494	6043	Chelseaview
224	225	fulan 225	Prefer not to say	75	651 Garden CourtApt. 769	4755	Muellerfurt
302	303	fulan 303	Prefer not to say	75	334 Olivia MeadowSuite 855	8400	North Paige
332	332	fulan 332	Male	75	011 Hamilton PlaceSuite 215	5702	East George
338	338	fulan 338	Female	75	899 Lynch LaneSuite 349	2652	Monahanberg
391	391	fulan 391	Male	75	4592 Isabella BoulevardApt. 793	7524	North Sophieland
409	409	fulan 409	Male	75	553 Alexander TrailSuite 694	134	New Chase

	customer_id	customer_name	gender	age	home_address	zip_code	city
439	439	fulan 439	Female	75	378 John MallSuite 835	3142	West Thomas
512	512	fulan 512	Prefer not to say	75	158 Joseph LaneApt. 252	3394	Clarkemouth
540	540	fulan 540	Prefer not to say	75	297 Ferry LaneApt. 061	8550	East Amelia
579	578	fulan 578	Prefer not to say	75	256 Andrew CrestApt. 193	8281	East Marcus
605	604	fulan 604	Prefer not to say	75	17 Connor ParadeApt. 442	4324	Conroymouth
629	628	fulan 628	Prefer not to say	75	774 Christiansen StreetSuite 261	2197	West Sophie
646	645	fulan 645	Female	75	988 Jacob CircuitSuite 385	1940	Greenfeldershire
963	957	fulan 957	Male	75	5215 Mitchell TrackSuite 778	4220	Samanthaberg
971	965	fulan 965	Prefer not to say	75	4627 Paige KnollApt. 635	5542	New Callumtown

In [37]:
```
customers_df.describe()
```

Out[37]:

	customer_id	age	zip_code
count	1001.000000	1001.000000	1001.000000
mean	500.942058	48.253746	5000.693307
std	289.013599	16.068215	2886.084454
min	1.000000	20.000000	2.000000
25%	251.000000	34.000000	2398.000000
50%	501.000000	50.000000	5079.000000
75%	751.000000	62.000000	7454.000000
max	1000.000000	75.000000	9998.000000

Membersihkan Data orders_df

In [40]:
```
datetime_columns = ["order_date", "delivery_date"]
for column in datetime_columns:
    orders_df[column] = pd.to_datetime(orders_df[column])
orders_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   order_id       1000 non-null   int64
 1   customer_id    1000 non-null   int64
 2   payment        1000 non-null   int64
 3   order_date     1000 non-null   datetime64[ns]
 4   delivery_date  1000 non-null   datetime64[ns]
dtypes: datetime64[ns](2), int64(3)
memory usage: 39.2 KB
Jumlah duplikasi:  6
```

In [41]:
```
product_df.drop_duplicates(inplace=True)
print("Jumlah duplikasi: ", product_df.duplicated().sum())
```

```
Jumlah duplikasi:  0
```

Membersihkan Data sales : 19 missing value pada kolom total_price.

In [42]:
```
sales_df[sales_df.total_price.isna()]
'''nilai total_price merupakan hasil perkalian antara price_per_unit dan quantity''
```

Out[42]:

	sales_id	order_id	product_id	price_per_unit	quantity	total_price
9	9	2	1196	105	1	NaN
121	121	27	1027	90	3	NaN
278	278	63	360	94	2	NaN
421	421	95	1091	115	1	NaN
489	489	108	1193	105	3	NaN
539	539	117	405	119	2	NaN
636	636	134	653	93	3	NaN
687	687	145	1138	102	1	NaN
854	854	177	64	104	1	NaN
1079	1079	222	908	94	3	NaN
1193	1193	248	1121	102	2	NaN
1313	1313	272	826	117	1	NaN
1548	1548	316	103	118	3	NaN
1688	1688	345	428	107	1	NaN
1775	1775	359	694	113	2	NaN
1902	1902	381	1218	105	3	NaN
2025	2025	408	611	112	3	NaN
2164	2164	436	583	100	3	NaN
2347	2347	476	696	113	2	NaN

In [43]:
```
sales_df["total_price"] = sales_df["price_per_unit"] * sales_df["quantity"]
```

In [44]:
```
sales_df.isna().sum()
```

Out[44]:
```
sales_id          0
order_id          0
product_id        0
price_per_unit    0
quantity          0
total_price       0
dtype: int64
```