

LAPORAN TUGAS BESAR PENGENALAN KOMPUTASI 2
ANALISIS DATASET *E-COMMERCE*
Pengenalan Komputasi KU1101

Oleh

Kelompok 4:

Audy Alicia Renatha Tirayoh/19623169
Nicholas Andhika Lucas/19623218
Amanda Nadine Calysta /16523204
Lintang Suminar/16523162
Reinhard Iven Winnata/16523071



SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JATINANGOR

NOVEMBER 2023

BAB I

PENDAHULUAN

1.1. Deskripsi data

Laporan ini menganalisis dataset yang berisi data aktivitas pembelian pelanggan dalam dataset *e-commerce* di Amerika milik suatu perusahaan, terbatas dari tanggal 1 – 9 November 2018. Dataset ini memberikan deskripsi berupa latar belakang pelanggan, pembelian yang dilakukan dan pembayarannya, serta proses pengiriman barang.

Platform *e-commerce* merekam sekumpulan data yang menjadi kunci pemahaman perilaku konsumen yang ditujukan untuk eksplorasi yang cermat terhadap dataset e-commerce Amerika yang komprehensif. Dalam laporan ini, diungkapkan analisis dari data mengenai aktivitas pembelian pelanggan selama satu tahun dalam dataset E-commerce Amerika dan bagaimana kebiasaan belanja online dari pelanggan

1.2. Informasi yang Ingin Diketahui

Informasi yang ingin diketahui dari dataset ini adalah analisis perilaku konsumen dari *E-Commerce* untuk memahami kebiasaan belanja online. Analisis perilaku ini dapat dilihat dari latar belakang pelanggan, jenis produk yang dibeli, dan proses pembayaran hingga pengiriman. Dengan demikian, dapat dicari korelasi dari suatu kebiasaan dengan kebiasaan lainnya.

1.3. Format data

Data yang digunakan merupakan data yang disimpan dalam format file *.csv* dengan nama *E-Com Dataset.csv*

1.4. Sumber data

Data yang digunakan ditemukan melalui *internet* dan diambil dari sumber: <https://www.kaggle.com/datasets/mervemenekse/ecommerce-dataset/data> yang dibuat oleh akun *Merve Menekse* pada tahun 2023.

1.5. Dimensi data

Data yang digunakan memiliki dimensi sebesar 2161x16 (baris dan kolom).

```
print("Dimensi data - baris:", df.shape[0], "- kolom:", df.shape[1])
✓ 0.0s
Dimensi data - baris: 2161 - kolom: 16
```

1.6. Ukuran data

Data yang digunakan memiliki ukuran file sebesar 224 KB.

1.7. Karakteristik data

a. Order_Date

Tanggal produk dipesan pelanggan.

Atribut ini berjenis kuantitatif numerik dengan range 2018-11-01 – 2018-11-09.

```
df["Order_Date"] = pd.to_datetime(df["Order_Date"])
max = df["Order_Date"].max().date()
min = df["Order_Date"].min().date()

print(f"Maximum date: {max}")
print(f"Minimum date: {min}")
Maximum date: 2018-11-09
Minimum date: 2018-11-01
```

b. Time

Waktu pemesanan produk dalam format ‘Hour:Minute:Second’

Atribut ini berjenis kuantitatif numerik dengan range 00:00:21 – 23:59:05.

```
df["Time"] = pd.to_datetime(df["Time"])
max = df["Time"].max().time()
min = df["Time"].min().time()

print(f"Maximum time: {max}")
print(f"Minimum time: {min}")
Maximum time: 23:59:05
Minimum time: 00:00:21
```

c. Aging

Waktu dari hari produk dipesan hingga hari pengiriman.

Atribut ini berjenis kuantitatif numerik dengan range 1–10.

```
max = df["Aging"].max()
min = df["Aging"].min()

print(f"Maximum aging: {max}")
print(f"Minimum aging: {min}")
Maximum aging: 10
Minimum aging: 1
```

d. Customer_id

ID unik yang dibuat untuk membedakan setiap pelanggan yang memesan.

Atribut ini berjenis kuantitatif numerik dengan range 10095–99991.

```
max = df["Customer_Id"].max()
min = df["Customer_Id"].min()
print(f"Maximum Customer ID: {max}")
print(f"Minimum Customer ID: {min}")
Maximum Customer ID: 99991
Minimum Customer ID: 10095
```

e. Gender

Jenis kelamin pelanggan.

Atribut ini berjenis kategorikal binary dan terdiri dari 2 pilihan kategori saja, yaitu Female dan Male.

```
print(df["Gender"].unique())
['Female' 'Male']
```

f. Device_Type

Jenis perangkat yang digunakan pelanggan untuk melakukan transaksi.

Atribut ini berjenis kategorikal binary dan terdiri dari 2 pilihan kategori saja, yaitu Web dan Mobile.

```
print(df["Device_Type"].unique())
['Web' 'Mobile']
```

g. Customer_Login_Type

Jenis saat melakukan transaksi seperti Member, Guest, dan First Signup.

Atribut ini berjenis kategorikal nominal yang berisi jenis login pelanggan tanpa urutan yang memiliki arti.

```
print(df["Customer_Login_type"].unique())
['Member' 'Guest' 'First SignUp']
```

h. Product_Category

Kategori dari produk yang dibeli pelanggan.

Atribut ini berjenis kategorikal nominal yang berisi kategori dari produk tanpa urutan yang memiliki arti.

```
print(df["Product_Category"].unique())
['Auto & Accessories' 'Fashion' 'Electronic' 'Home & Furniture']
```

i. Product

Jenis produk beragam sesuai dengan kategori produk yang dibeli pelanggan.

Atribut ini berjenis kategorikal nominal yang berisi jenis-jenis kategori tanpa urutan yang memiliki arti.

```
print(df["Product"].unique())
['Car Body Covers' 'Car Mat' 'Tyre' 'Car Pillow & Neck Rest'
 'Car Speakers' 'Car Media Players' 'Bike Tyres' 'Car & Bike
Care'
 'Car Seat Covers' 'Formal Shoes' 'Casula Shoes' 'Suits' 'Shirts'
 'T - Shirts' 'Running Shoes' 'Jeans' 'Titak watch' 'Fossil
Watch'
 'Sneakers' 'Sports Wear' 'LED' 'Watch' 'Keyboard' 'Samsung
Mobile'
 'Tablet' 'LCD' 'Speakers' 'Mixer/Juicer' 'Apple Laptop' 'Sofas'
 'Bed Sheets' 'Towels' 'Umbrellas' 'Curtains' 'Beds' 'Shoe Rack'
 'Sofa Covers' 'Dinning Tables' 'Dinner Crockery' 'Fans' 'Mouse'
 'Iron']
```

j. Sales

Total jumlah penjualan dari produk yang dibeli pelanggan.

Atribut ini berjenis kuantitatif numerik dengan range 33–250.

```
max = df["Sales"].max()
min = df["Sales"].min()

print(f"Maximum Sales: {max}")
print(f"Minimum Sales: {min}")
Maximum Sales: 250
Minimum Sales: 33
```

k. Quantity

Jumlah unit produk yang dipesan pelanggan.

Atribut ini berjenis kuantitatif numerik dengan range 1–5.

```
max = df["Quantity"].max()
min = df["Quantity"].min()

print(f"Maximum Quantity: {max}")
print(f"Minimum Quantity: {min}")
Maximum Quantity: 5
Minimum Quantity: 1
```

l. Discount

Persentase tingkat diskon yang didapatkan pelanggan.

Atribut ini berjenis kuantitatif numerik dengan range $0.1 - 0.5$.

```
max = df["Discount"].max()
min = df["Discount"].min()

print(f"Maximum Discount: {max}")
print(f"Minimum Discount: {min}")
Maximum Discount: 0.5
Minimum Discount: 0.1
```

m. Profit

Keuntungan yang diperoleh pelanggan setelah diskon diaplikasikan pada pemesanannya.

Atribut ini berjenis kuantitatif numerik dengan range $0.5 - 167.5$

```
max = df["Profit"].max()
min = df["Profit"].min()

print(f"Maximum Profit: {max}")
print(f"Minimum Profit: {min}")
Maximum Profit: 167.5
Minimum Profit: 0.5
```

n. Shipping_cost

Biaya pengiriman dari produk yang dibeli pelanggan.

Atribut ini berjenis kuantitatif numerik dengan range $0.1 - 16.8$.

```
max = df["Shipping_Cost"].max()
min = df["Shipping_Cost"].min()

print(f"Maximum Shipping Cost: {max}")
print(f"Minimum Shipping Cost: {min}")
Maximum Shipping Cost: 16.8
Minimum Shipping Cost: 0.1
```

o. Order_Priority

Prioritas pengiriman pesanan yang dipilih pelanggan berupa critical, medium, high, dan low.

Atribut ini berjenis ordinal karena memiliki urutan yang mempunyai arti atau ranking.

```
print(df["Order_Priority"].unique())
['Critical' 'Medium' 'High' 'Low']
```

p. Payment_method

Metode pembayaran yang digunakan pelanggan berupa e-wallet, money order, dan credit card.

Atribut ini berjenis kategorikal nominal yang berisi metode pembayaran yang digunakan tanpa urutan yang memiliki arti.

```
print(df["Payment_method"].unique())
['credit_card' 'money_order' 'debit_card' 'e_wallet']
```

BAB II

PEMBERSIHAN DATA

Data sampel yang dipilih ternyata terdeteksi memiliki nilai yang *null* atau *Not Applicable* (NA) sehingga perlu ada tata cara untuk membersihkan data. *Data cleansing* umumnya dieksekusi sebelum data diolah maupun dikarakterisasi. Fungsi dari *data cleansing* diletakkan di awal pengkodean supaya data terdefinisi secara jelas di pengolahan maupun di bagian karakterisasi. Data perlu dibersihkan dari data - data yang tidak sah agar pengolahan data valid dan efisien. Berikut beberapa cara untuk dapat mendeteksi dan menghilangkan data - data yang sekiranya memiliki *null value* di baris atau kolom tertentu.

2.1 Mendeteksi data yang null

```
df.isnull().sum()
[1]:
```

Order_Date	0
Time	0
Aging	1
Customer_Id	0
Gender	0
Device_Type	0
Customer_Login_type	0
Product_Category	0
Product	0
Sales	1
Quantity	2
Discount	1
Profit	0
Shipping_Cost	1
Order_Priority	2
Payment_method	0
dtype: int64	

Data yang *null* diperiksa dengan memanggil fungsi *isnull()* dan dijumlahkan per kolom *datasheet* dengan fungsi *sum()*. Hasil dari pemanggilan fungsi *isnull().sum()* adalah tabel berupa jumlah data yang memiliki nilai *null* dengan masing - masing kolom *datasheet* dari kiri ke kanan

2.2 Memeriksa data yang *Not applicable*

```
df.isna().sum()
[1]:
```

Order_Date	0
Time	0
Aging	1
Customer_Id	0
Gender	0
Device_Type	0
Customer_Login_type	0
Product_Category	0
Product	0
Sales	1
Quantity	2
Discount	1
Profit	0
Shipping_Cost	1
Order_Priority	2
Payment_method	0
dtype: int64	

Data yang *not applicable* diperiksa dengan memanggil fungsi *isna()* dan dijumlahkan per kolom *datasheet* dengan fungsi *sum()*. Hasil dari pemanggilan fungsi *isna().sum()* adalah tabel berupa jumlah data yang memiliki nilai *not applicable* dengan masing - masing kolom *datasheet* dari kiri ke kanan.

2.3 Memeriksa data yang *Not applicable*

```
> ^ #Mengeluarkan info yang null
  print(df.info())
[47]
...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Order_Date      51290 non-null   object  
 1   Time            51290 non-null   object  
 2   Aging           51289 non-null   float64 
 3   Customer_Id    51290 non-null   int64  
 4   Gender          51290 non-null   object  
 5   Device_Type    51290 non-null   object  
 6   Customer_Login_type 51290 non-null   object  
 7   Product_Category 51290 non-null   object  
 8   Product         51290 non-null   object  
 9   Sales           51289 non-null   float64 
 10  Quantity        51288 non-null   float64 
 11  Discount        51289 non-null   float64 
 12  Profit          51290 non-null   float64 
 13  Shipping_Cost   51289 non-null   float64 
 14  Order_Priority  51288 non-null   object  
 15  Payment_method   51290 non-null   object  
dtypes: float64(6), int64(1), object(9)
memory usage: 6.3+ MB
None
```

Data yang bernilai *not applicable* atau *null* ditampilkan bagian dan jenisnya dengan memanggil fungsi *info()*. Fungsi ini dipanggil sehingga terlihat berapa jumlah objek yang ada di dalam kolom tertentu yang bernilai tidak *not applicable* atau *null*.

2.4 Mengumpulkan bagian data yang memiliki nilai *null* atau *not applicable*

```
# Mengumpulkan data yang bernilai null
null_data = df[df.isnull().any(axis=1)]

# Menampilkan hasil
print(null_data)

   Order_Date      Time  Aging  Customer_Id  Gender Device_Type \
27      5/2/2018  11:45:38    NaN       26058  Female        Web
95     4/22/2018  11:32:22    5.0       52267   Male        Web
211    8/5/2018  17:27:54    6.0       47137   Male        Web
321    6/5/2018  11:04:11    3.0       41850   Male        Web
535    4/16/2018  16:20:02    3.0       13777   Male        Web
625   10/15/2018  20:16:34    2.0       26367   Male        Web
791    7/3/2018  23:40:16    4.0       36902  Female        Web
793   5/16/2018  21:30:59    6.0       16381   Male        Web

  Customer_Login_type  Product_Category          Product  Sales \
27            Member  Auto & Accessories  Car Media Players  140.0
95            Member  Auto & Accessories      Bike Tyres   72.0
211           Member  Auto & Accessories        Tyre Mat  250.0
321           Member  Auto & Accessories        Car Mat   54.0
535           Member  Auto & Accessories        Tyre Mat  250.0
625           Member  Auto & Accessories        Tyre Mat  250.0
791           Member  Auto & Accessories  Car Pillow & Neck Rest  231.0
793           Member  Auto & Accessories      Car Speakers    NaN

  Quantity  Discount  Profit  Shipping_Cost Order_Priority Payment_method
27      1.0      0.3   55.8         5.6        High credit_card
95     NaN      0.1    36.0         3.6    Critical credit_card
211     5.0     NaN  132.5        13.3     Medium credit_card
321     NaN      0.2    54.0         5.4    Critical credit_card
535     4.0      0.2   150.0        NaN    Critical credit_card
625     4.0      0.3   140.0        14.0     NaN debit_card
791     1.0      0.1   148.7        14.9     NaN money_order
793     1.0      0.1   124.7        12.5    Critical credit_card
```

Data yang bernilai *not applicable* atau *null* dikumpulkan dan diseleksi dengan memanggil fungsi `isnull().any(axis=1)` kemudian ditampilkan hasil data tersebut. Data yang memiliki nilai `NaN` data tersebut tidak dapat digunakan dan akan dibersihkan di tahap selanjutnya.

2.5 Membersihkan data yang bernilai *null* atau *not applicable*

```
#Membersihkan data yang kosong
df.dropna(inplace = True)
```

Data yang bernilai *not applicable* atau *null* dihilangkan dengan memanggil fungsi `dropna(inplace = True)`. Memanggil fungsi ini berarti menghilangkan tahap untuk pengolahan data yang bernilai *not applicable* atau *null*.

2.6 Memverifikasi data sudah bersih dari nilai *not applicable* atau *null*

```
#Memverifikasi data sudah clean
df.isna().sum()

Order_Date      0
Time            0
Aging           0
Customer_Id    0
Gender          0
Device_Type    0
Customer_Login_type 0
Product_Category 0
Product         0
Sales            0
Quantity        0
Discount        0
Profit           0
Shipping_Cost   0
Order_Priority  0
Payment_Method  0
dtype: int64
```

Data yang sudah dibersihkan harus divalidasi kembali. Syarat data dikatakan bersih apabila nilai dari pemanggilan fungsi *isna().sum()* bernilai nol untuk semua kolom yang terdapat di *data sheet*. Jika syarat ini terpenuhi maka data berhasil dibersihkan.

BAB III

HASIL ANALISIS

3.1. Sampel data

Sampel data yang diambil adalah **62 data pertama**, yaitu sekitar 3% dari populasi.

1	Order_Dat	Time	Aging	Customer_Gender	Device_Typ	Customer	Product_C	Product	Sales	Quantity	Discount	Profit	Shipping_C	Order_Pric	Payment_method
2	11/1/2018	18:52:12	5	31345 Female	Web	Member	Auto & Acc	Car Body C	117	1	0.2	32.3	3.2	Critical	credit_card
3	11/1/2018	10:48:19	10	18692 Male	Web	Member	Auto & Acc	Car Mat	54	1	0.3	54	5.4	Medium	credit_card
4	11/1/2018	14:58:41	2	40363 Male	Web	Member	Auto & Acc	Tyre	250	1	0.2	160	16	Critical	credit_card
5	11/1/2018	16:54:38	1	24997 Male	Web	Member	Auto & Acc	Car Pillow i	231	4	0.2	132.5	13.3	Medium	money_order
6	11/1/2018	17:57:54	5	13491 Male	Web	Member	Auto & Acc	Car Body C	117	4	0.3	18.3	1.8	High	credit_card
7	11/1/2018	16:37:54	4	15219 Male	Web	Member	Auto & Acc	Car Speake	211	1	0.3	105.7	10.6	Critical	credit_card
8	11/1/2018	13:30:12	2	43804 Male	Web	Member	Auto & Acc	Car Body C	117	1	0.2	34.7	3.5	Critical	credit_card
9	11/1/2018	19:18:06	2	58913 Male	Web	Member	Auto & Acc	Car Pillow i	231	4	0.3	123.3	12.3	High	credit_card
10	11/1/2018	0:22:08	9	27917 Male	Web	Member	Auto & Acc	Car Media	140	1	0.3	47.4	4.7	Critical	debit_card
11	11/1/2018	17:02:45	6	46398 Male	Web	Member	Auto & Acc	Car Media	140	1	0.1	55.8	5.6	High	credit_card
12	11/1/2018	14:42:31	8	50045 Male	Web	Member	Auto & Acc	Car Body C	117	1	0.3	25.3	2.5	High	credit_card
13	11/1/2018	23:35:06	4	54319 Male	Web	Member	Auto & Acc	Tyre	250	1	0.2	160	16	High	credit_card
14	11/1/2018	21:23:20	4	25312 Male	Web	Member	Auto & Acc	Bike Tyres	72	1	0.3	36	3.6	High	credit_card
15	11/1/2018	22:25:27	3	52156 Male	Web	Member	Auto & Acc	Car Mat	54	1	0.3	54	5.4	High	money_order
16	11/1/2018	21:22:53	5	31737 Male	Web	Member	Auto & Acc	Car & Bike	118	1	0.3	27.4	2.7	Medium	credit_card
17	11/1/2018	21:22:16	4	27346 Male	Web	Member	Auto & Acc	Car Mat	54	1	0.3	18	1.8	Critical	credit_card
18	11/1/2018	0:54:05	4	12941 Male	Web	Member	Auto & Acc	Car Body C	117	1	0.3	25.3	2.5	High	credit_card
19	11/1/2018	13:52:56	10	18379 Male	Web	Member	Auto & Acc	Car Media	140	1	0.3	51.6	5.2	Critical	money_order
20	11/1/2018	23:24:08	1	40559 Male	Web	Member	Auto & Acc	Car Seat Cc	114	4	0.1	29.4	2.9	High	credit_card
21	11/1/2018	21:27:25	10	11218 Male	Web	Member	Auto & Acc	Car Speak	211	1	0.1	124.7	12.5	High	money_order
22	11/1/2018	10:06:37	6	26129 Female	Web	Member	Auto & Acc	Car Seat Cc	114	1	0.2	27.2	2.7	High	credit_card
23	11/1/2018	23:28:16	4	30800 Male	Web	Member	Auto & Acc	Car & Bike	118	1	0.3	23.8	2.4	Medium	credit_card
24	11/1/2018	7:19:59	10	32966 Female	Web	Member	Auto & Acc	Tyre	250	4	0.3	120	12	High	credit_card
25	11/1/2018	14:44:38	5	59771 Male	Web	Guest	Auto & Acc	Car & Bike	118	5	0.3	20.3	2	Medium	credit_card
26	11/1/2018	0:02:01	8	51330 Male	Web	Member	Auto & Acc	Car Pillow i	231	4	0.3	104.8	10.5	High	credit_card
27	11/1/2018	16:59:05	2	39633 Female	Web	Member	Auto & Acc	Car Mat	54	1	0.3	54	5.4	Medium	credit_card
28	11/1/2018	20:44:34	5	54109 Male	Web	Member	Auto & Acc	Car Seat Cc	114	1	0.3	30.6	3.1	Medium	credit_card
29	11/1/2018	13:30:49	5	23088 Male	Web	Member	Auto & Acc	Pillow i	231	1	0.2	137.1	13.7	High	credit_card
30	11/1/2018	23:01:03	7	40124 Male	Web	Member	Auto & Acc	Car Media	140	4	0.2	48.8	4.9	Medium	money_order
31	11/1/2018	21:16:43	3	50072 Male	Web	Member	Fashion	Formal Sh	213	1	0.4	124.5	12.4	Medium	credit_card
32	11/1/2018	0:29:18	3	45817 Male	Web	Member	Fashion	Casula Sho	122	5	0.5	35.9	3.6	Medium	credit_card
33	11/1/2018	22:14:52	6	46926 Male	Web	Member	Fashion	Casula Sho	122	1	0.4	22.5	2.2	Critical	credit_card
34	11/1/2018	7:31:32	10	13573 Female	Web	Member	Fashion	Suits	109	2	0.5	26.8	2.7	High	money_order
35	11/1/2018	13:56:24	1	19287 Female	Web	Member	Fashion	Shirts	196	3	0.5	110.1	11	Medium	credit_card
36	11/1/2018	8:22:05	1	37477 Male	Web	Member	Fashion	T - Shirts	248	5	0.5	155.6	15.6	Medium	credit_card
37	11/1/2018	18:33:19	9	55395 Female	Web	Member	Fashion	Suits	109	1	0.5	24.6	2.5	High	e_wallet
38	11/1/2018	20:25:36	7	59234 Male	Web	Member	Fashion	Running Sh	224	3	0.5	137.3	13.7	Medium	credit_card
39	11/1/2018	13:39:53	1	29532 Male	Web	Member	Fashion	Jeans	218	3	0.4	111.8	11.2	Critical	credit_card
40	11/1/2018	12:50:41	1	13490 Female	Web	Member	Fashion	Titak watch	228	2	0.4	129.8	13	Medium	credit_card
41	11/1/2018	12:57:54	1	17239 Male	Web	Member	Fashion	Jeans	218	1	0.4	133.6	13.4	High	credit_card
42	11/1/2018	13:42:49	1	30561 Male	Web	Member	Fashion	Shirts	196	5	0.5	106.2	10.6	Medium	credit_card
43	11/1/2018	19:00:07	1	34886 Male	Web	Member	Fashion	Fossil Watt	159	5	0.5	71.1	7.1	High	credit_card
44	11/1/2018	8:06:08	1	18929 Male	Web	Member	Fashion	Running Sh	224	3	0.5	123.8	12.4	Medium	credit_card
45	11/1/2018	16:54:09	1	17360 Male	Web	Member	Fashion	Casula Sho	122	5	0.5	35.9	3.6	Critical	money_order
46	11/1/2018	18:38:31	1	35348 Male	Web	Member	Fashion	Jeans	218	3	0.4	111.8	11.2	Critical	credit_card
47	11/1/2018	21:26:22	1	47139 Male	Web	Member	Fashion	Sneakers	62	1	0.4	62	6.2	High	credit_card
48	11/1/2018	19:34:51	1	45362 Female	Web	Member	Fashion	Fossil Watt	159	1	0.5	47.2	4.7	High	credit_card
49	11/1/2018	23:27:57	1	59032 Female	Web	Member	Fashion	Casula Sho	122	5	0.4	29.8	3	Critical	credit_card
50	11/1/2018	14:59:16	1	41958 Male	Web	Member	Fashion	Suits	109	2	0.4	24.6	2.5	High	money_order
51	11/1/2018	10:28:27	1	22522 Male	Web	Member	Fashion	Titak watch	228	2	0.5	134.3	13.4	Critical	credit_card
52	11/1/2018	19:40:51	1	10883 Male	Web	Member	Fashion	Formal Sh	213	1	0.4	124.5	12.4	Medium	money_order
53	11/1/2018	0:03:15	1	10623 Female	Web	Member	Fashion	Formal Sh	213	5	0.5	122.4	12.2	Medium	credit_card
54	11/1/2018	10:03:19	1	31998 Female	Web	Member	Fashion	Shirts	196	5	0.5	67	6.7	High	credit_card
55	11/1/2018	12:37:20	1	40438 Male	Web	Member	Fashion	Sports We	85	2	0.5	3.3	0.3	Medium	money_order
56	11/1/2018	13:30:33	1	14450 Male	Web	Member	Fashion	Sneakers	62	5	0.5	12.4	1.2	Medium	money_order
57	11/1/2018	14:11:25	3	35425 Male	Web	Member	Fashion	Fossil Watt	159	4	0.3	59.9	6	High	credit_card
58	11/1/2018	23:32:19	5	33910 Male	Web	Member	Fashion	Fossil Watt	159	4	0.3	59.9	6	Critical	credit_card
59	11/1/2018	14:50:39	3	51622 Male	Web	Member	Fashion	Casula Sho	122	2	0.5	29.8	3	Medium	credit_card
60	11/1/2018	18:24:45	7	14641 Male	Web	Member	Fashion	Formal Sh	213	2	0.4	116	11.6	Medium	credit_card
61	11/1/2018	22:05:17	9	44317 Female	Web	Member	Fashion	Titak watch	228	1	0.4	143.4	14.3	Medium	credit_card
62	11/1/2018	21:33:30	3	34127 Female	Web	Member	Fashion	T - Shirts	248	4	0.3	138.2	13.8	Critical	credit_card
63	11/1/2018	13:47:04	9	40591 Female	Web	Guest	Fashion	Titak watch	228	4	0.5	138.9	13.9	High	credit_card

3.2. Statistik data

Karena ukuran data yang besar, untuk menentukan segala *stat* atau statistik dari data tersebut, akan dilakukan proses analisis menggunakan *python*, sehingga didapatkan beberapa unsur statistik dari data yang digunakan, yaitu:

1. Rata-rata (mean)
Rata-rata nilai dari sebuah variabel tertentu
2. Standar Deviasi (std)
Besaran nilai standar deviasi untuk setiap kolom
3. Persentil (10%, 25%, 50%, 75%, 90%)
Suatu ukuran atau nilai yang membagi distribusi frekuensi dari data menjadi 100 bagian yang sama besar. Pada analisis statistik ini, dicari persentil pada titik 10%, 25%, 50%, 75%, dan 90%.
4. Ekstremum (nilai maksimum dan minimum)
Nilai terbesar dan terkecil dari setiap kolom
5. Distribusi frekuensi nilai
Persebaran frekuensi dari suatu nilai yang terdapat dalam suatu kolom.

A. Rata-Rata, Standar Deviasi, dan Ekstremum

```
sample_df.describe()
✓ 0.0s
```

	Aging	Customer_Id	Sales	Quantity	Discount	Profit	Shipping_Cost
count	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000
mean	4.000000	33988.145161	160.596774	2.451613	0.354839	75.116129	7.509677
std	3.078215	14633.821501	61.679887	1.595870	0.118290	48.179317	4.819928
min	1.000000	10623.000000	54.000000	1.000000	0.100000	3.300000	0.300000
25%	1.000000	20095.750000	117.000000	1.000000	0.300000	29.800000	3.000000
50%	3.000000	34018.500000	149.500000	2.000000	0.300000	57.850000	5.800000
75%	6.000000	45703.250000	218.000000	4.000000	0.500000	123.675000	12.375000
max	10.000000	59771.000000	250.000000	5.000000	0.500000	160.000000	16.000000

Menggunakan fungsi statistik sederhana `describe()`, didapatkan nilai rata-rata (mean), standar deviasi (std), nilai tertinggi (max), dan nilai terendah (min) dari 62 data pertama dataset *E-Commerce*, dengan `count` adalah banyak data yang terhitung sesuai dengan sampel yang diambil.

B. Persentil (10%, 25%, 50%, 75%, 90%)

```
# Hanya memilih kolom berjenis numerik
kolom_numerik = sample_df.select_dtypes(include=['number'])

# Menghitung persentil sampel
kolom_numerik.quantile([0.1, 0.25, 0.5, 0.75, 0.9])
```

✓ 0.0s

	Aging	Customer_Id	Sales	Quantity	Discount	Profit	Shipping_Cost
0.10	1.0	13660.70	73.3	1.0	0.2	23.880	2.410
0.25	1.0	20095.75	117.0	1.0	0.3	29.800	3.000
0.50	3.0	34018.50	149.5	2.0	0.3	57.850	5.800
0.75	6.0	45703.25	218.0	4.0	0.5	123.675	12.375
0.90	9.0	53913.70	231.0	5.0	0.5	137.280	13.700

Menggunakan fungsi statistik quantile(), dapat ditentukan persentil pada titik 10%, 25%, 50%, 75%, dan 90%.

C. Distribusi Frekuensi Nilai

1. Gender

```
print(sample_df["Gender"].value_counts())
✓ 0.0s
```

Gender	count
Male	47
Female	15

Name: count, dtype: int64

2. Device Type

```
print(sample_df["Device_Type"].value_counts())
✓ 0.0s
```

Device_Type	count
Web	62

Name: count, dtype: int64

3. Customer Login Type

```
print(sample_df["Device_Type"].value_counts())
✓ 0.0s
```

Device_Type	count
Web	62

Name: count, dtype: int64

4. Product Category

```
print(sample_df["Product_Category"].value_counts())
✓ 0.0s

Product_Category
Fashion           33
Auto & Accessories   29
Name: count, dtype: int64
```

5. Product

Product			
Car Body Covers	5	Shirts	3
Casula Shoes	5	Suits	3
Car Pillow & Neck Rest	4	Car Seat Covers	3
Fossil Watch	4	Car & Bike Care	3
Car Media Players	4	T - Shirts	2
Titak watch	4	Running Shoes	2
Formal Shoes	4	Car Speakers	2
Car Mat	4	Sneakers	2
Tyre	3	Bike Tyres	1
Jeans	3	Sports Wear	1
		Name: count, dtype: int64	

6. Order Priority

```
print(sample_df["Order_Priority"].value_counts())
✓ 0.0s

Order_Priority
High      24
Medium    23
Critical  15
Name: count, dtype: int64
```

7. Payment Method

```
print(sample_df["Payment_method"].value_counts())
✓ 0.0s

Payment_method
credit_card   49
money_order   11
debit_card    1
e_wallet      1
Name: count, dtype: int64
```

D. Total dan Rata-Rata Penjualan

```

total_terbesar = 0
for i in set(sample_df["Product_Category"]):
    df1 = sample_df.loc[sample_df["Product_Category"] == i]
    total_sales = df1["Sales"].sum()
    if total_sales > total_terbesar:
        total_terbesar = total_sales
        j = i
    print(f"Total Penjualan Kategori {i} adalah {total_sales}")
print()
print(f"Total penjualan terbesar adalah {total_terbesar} dari kategori {j}")
✓ 0.0s

Total Penjualan Kategori Auto & Accessories adalah 4225
Total Penjualan Kategori Fashion adalah 5732

Total penjualan terbesar adalah 5732 dari kategori Fashion

average_terbesar = 0
for i in set(sample_df["Product_Category"]):
    df1 = sample_df.loc[sample_df["Product_Category"] == i]
    average_sales = df1["Sales"].mean()
    if average_sales > average_terbesar:
        average_terbesar= average_sales
        j = i
    print(f"Rata-Rata Penjualan Kategori {i} adalah {average_sales:.2f}")
print()
print(f"Rata-rata penjualan terbesar adalah {average_terbesar:.2f} dari kategori {j}")
✓ 0.0s

Rata-Rata Penjualan Kategori Auto & Accessories adalah 145.69
Rata-Rata Penjualan Kategori Fashion adalah 173.70

Rata-rata penjualan terbesar adalah 173.70 dari kategori Fashion

```

Data di atas menunjukkan rata-rata (mean) dan total (sum) dari penjualan setiap produk kategori pada tanggal 1–9 November 2018 dari sampel data yang ada. Data tersebut menunjukkan penjualan terbanyak dari produk kategori *Fashion* dengan total penjualan 5732 produk dan rata-rata penjualan terbesar dari produk kategori *Fashion* dengan rata-rata penjualan dari seluruh kategori pada bulan November 2018 sebesar 173.70 produk.

E. Total dan Rata-Rata Kuantitas Pembelian per Produk

```

total_terbesar = 0
for i in set(sample_df["Product"]):
    df1 = sample_df.loc[sample_df["Product"] == i]
    total_quantity = df1["Quantity"].sum()
    if total_quantity > total_terbesar:
        total_terbesar = total_quantity
        j = i
    print(f"Total kuantitas pembelian produk {i} adalah {total_quantity}")
print()
print(f"Total kuantitas pembelian adalah {total_terbesar} berupa produk {j}")

```

```

average_terbesar = 0
for i in set(sample_df["Product"]):
    df1 = sample_df.loc[sample_df["Product"] == i]
    average_quantity = df1["Quantity"].mean()
    if average_quantity > average_terbesar:
        average_terbesar= average_quantity
        j = i
    print(f"Rata-Rata kuantitas pembelian produk {i} adalah {average_quantity:.2f}")
print()
print(f"Rata-rata kuantitas pembelian terbesar adalah {average_terbesar:.2f} berupa produk {j}")

```

Untuk mencari total dan rata-rata kuantitas pembelian per-*product* dan total penjualan dan rata-rata terbesar sekaligus nama *product*-nya, digunakan algoritma di atas.

Dari algoritma tersebut didapatkan data sebagai berikut:

```

Total kuantitas pembelian produk Car Body Covers adalah 8
Total kuantitas pembelian produk Casula Shoes adalah 18
Total kuantitas pembelian produk Car Mat adalah 4
Total kuantitas pembelian produk Running Shoes adalah 6
Total kuantitas pembelian produk Car Seat Covers adalah 6
Total kuantitas pembelian produk Formal Shoes adalah 9
Total kuantitas pembelian produk Car Pillow & Neck Rest adalah 13
Total kuantitas pembelian produk Car Speakers adalah 2
Total kuantitas pembelian produk Suits adalah 5
Total kuantitas pembelian produk Tyre adalah 6
Total kuantitas pembelian produk Car Media Players adalah 7
Total kuantitas pembelian produk Sneakers adalah 6
Total kuantitas pembelian produk Fossil Watch adalah 14
Total kuantitas pembelian produk Sports Wear adalah 2
Total kuantitas pembelian produk Car & Bike Care adalah 7
Total kuantitas pembelian produk Titak watch adalah 9
Total kuantitas pembelian produk T - Shirts adalah 9
Total kuantitas pembelian produk Jeans adalah 7
Total kuantitas pembelian produk Bike Tyres adalah 1
Total kuantitas pembelian produk Shirts adalah 13

Total kuantitas pembelian adalah 18 berupa produk Casula Shoes

```

```

Rata-Rata kuantitas pembelian produk Car Body Covers adalah 1.60
Rata-Rata kuantitas pembelian produk Casula Shoes adalah 3.60
Rata-Rata kuantitas pembelian produk Car Mat adalah 1.00
Rata-Rata kuantitas pembelian produk Running Shoes adalah 3.00
Rata-Rata kuantitas pembelian produk Car Seat Covers adalah 2.00
Rata-Rata kuantitas pembelian produk Formal Shoes adalah 2.25
Rata-Rata kuantitas pembelian produk Car Pillow & Neck Rest adalah 3.25
Rata-Rata kuantitas pembelian produk Car Speakers adalah 1.00
Rata-Rata kuantitas pembelian produk Suits adalah 1.67
Rata-Rata kuantitas pembelian produk Tyre adalah 2.00
Rata-Rata kuantitas pembelian produk Car Media Players adalah 1.75
Rata-Rata kuantitas pembelian produk Sneakers adalah 3.00
Rata-Rata kuantitas pembelian produk Fossil Watch adalah 3.50
Rata-Rata kuantitas pembelian produk Sports Wear adalah 2.00
Rata-Rata kuantitas pembelian produk Car & Bike Care adalah 2.33
Rata-Rata kuantitas pembelian produk Titak watch adalah 2.25
Rata-Rata kuantitas pembelian produk T - Shirts adalah 4.50
Rata-Rata kuantitas pembelian produk Jeans adalah 2.33
Rata-Rata kuantitas pembelian produk Bike Tyres adalah 1.00
Rata-Rata kuantitas pembelian produk Shirts adalah 4.33

Rata-rata kuantitas pembelian terbesar adalah 4.50 berupa produk T - Shirts

```

Data di atas menunjukkan rata-rata (mean) dan total (sum) dari penjualan setiap produk kategori pada bulan November 2018. Data tersebut menunjukkan kuantitas pembelian terbanyak dari produk *Casula Shoes* dengan total kuantitas pembelian 18 produk dan rata-rata kuantitas pembelian terbesar dari produk *T-Shirts* dengan rata-rata kuantitas pembelian dari seluruh produk pada bulan November 2018 sebesar 4,50 produk.

3.3. Visualisasi

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.dates as mdates
df = pd.read_csv("E-Com Dataset.csv")

```

Untuk mempermudah gambaran hubungan antar atribut-atribut, dilakukan visualisasi data menggunakan *Python* melalui *matplotlib*. Kami juga menggunakan *seaborn* untuk menjadikan grafik statistik yang kami buat dapat terlihat lebih menarik dan informatif.

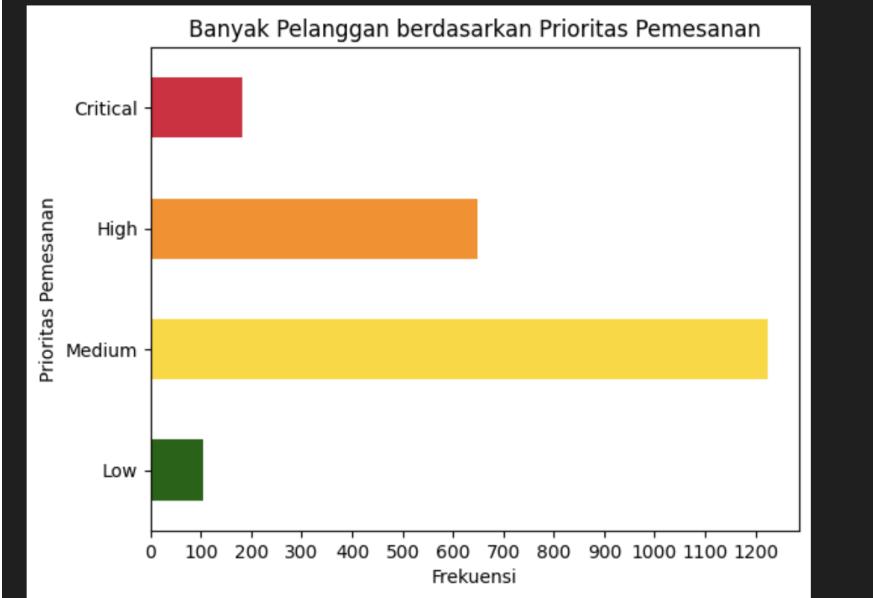
Berikut hasil yang didapat dari visualisasi data

A. Perbandingan Kategori

1. Diagram Batang Horizontal dari Banyak Pelanggan berdasarkan Prioritas Pemesanan

```
# Membuat list untuk jenis kategori yang ada dalam prioritas order
kategori = ["Low", "Medium", "High", "Critical"]
frekuensi = df["Order_Priority"].value_counts().loc[kategori]

frekuensi.plot(kind="barh", color=['darkgreen', 'gold', 'darkorange', 'crimson'])
plt.title("Banyak Pelanggan berdasarkan Prioritas Pemesanan")
plt.xlabel("Frekuensi")
plt.xticks(range(0, max(frekuensi), 100))
plt.ylabel("Prioritas Pemesanan")
plt.show()
```



Insight: Prioritas pemesanan yang paling banyak dipilih adalah medium, diikuti dengan high, critical, lalu low.

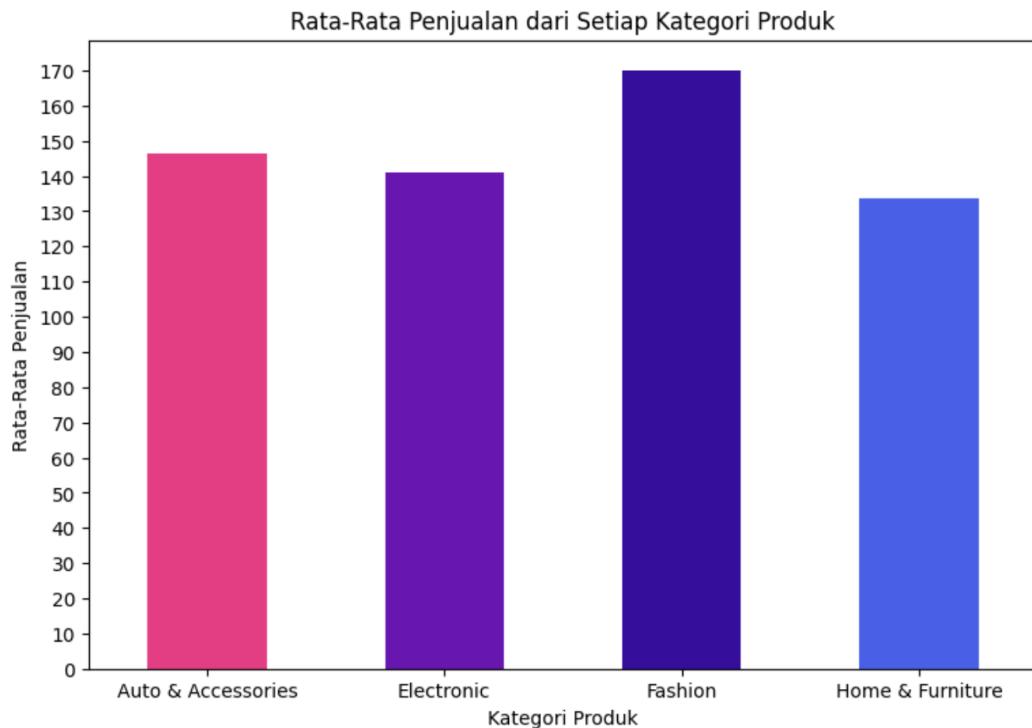
Pemilihan warna:

- Merah: menandai prioritas critical (sangat tinggi)
- Oranye: menandai prioritas high (tinggi)
- Kuning: menandai prioritas medium (menengah)
- Hijau: menandai prioritas low (rendah)

2. Diagram Batang Vertikal Rata-Rata Penjualan dari Setiap Kategori Produk

```
average_sales_by_category = df.groupby('Product_Category')['Sales'].mean()

plt.figure(figsize=(9, 6))
average_sales_by_category.plot(kind='bar', color= ['#F72585", "#7209B7", "#3A0CA3", "#4361EE", "#4CC9F0"])
plt.title('Rata-Rata Penjualan dari Setiap Kategori Produk')
plt.xlabel('Kategori Produk')
plt.xticks(rotation=0)
plt.ylabel('Rata-Rata Penjualan')
plt.yticks()
plt.ylim(0, int(max(average_sales_by_category)) + 10, 10)
plt.show()
```



Insight: Kategori produk dengan penjualan rata-rata terbanyak adalah kategori Fashion, diikuti Auto & Accessories, Electronic, dan Home & Furniture.

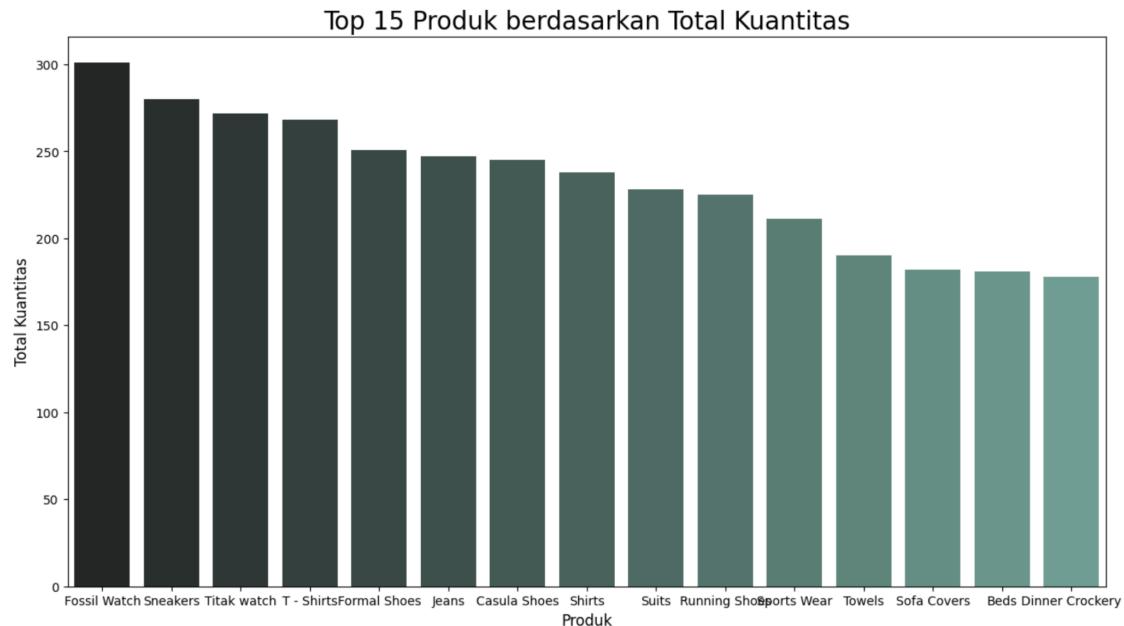
Pemilihan warna:

- Ungu: menandai produk dengan kategori elektronik (*Electronic*)
- Biru: menandai produk dengan kategori perabot rumah tangga (*Home & Furniture*)
- Biru Tua: menandai produk dengan kategori mode (*Fashion*)
- Pink: menandai produk dengan kategori otomotif dan aksesorisnya (*Auto & Accessories*)

3. Diagram Batang Horizontal Top 15 Produk berdasarkan Total Kuantitas

```
total_quantity = df.groupby('Product')['Quantity'].sum().reset_index()
total_quantity_sorted = total_quantity.sort_values(by='Quantity', ascending=False)
top_15_produk = total_quantity_sorted.head(15)

plt.figure(figsize=(15, 8))
sns.barplot(x='Produk', y='Quantity', data=top_15_produk, palette="dark:#5A9")
plt.xlabel('Produk', fontsize = 12)
plt.ylabel('Total Kuantitas', fontsize = 12)
plt.title('Top 15 Produk berdasarkan Total Kuantitas', fontsize = 20)
plt.show()
```



Insight: Produk dengan kuantitas terbanyak adalah Fossil Watch.

Pemilihan warna:

Semakin gelap warna batang, maka semakin tinggi kuantitas pembelian produk dari setiap produk.

- Diagram Batang Bertingkat Banyak Pelanggan dari Setiap Jenis Kelamin dan Metode Pembayaran

```

grouped_data = df.groupby(["Gender", "Payment_method"])["Customer_Id"].size().unstack()

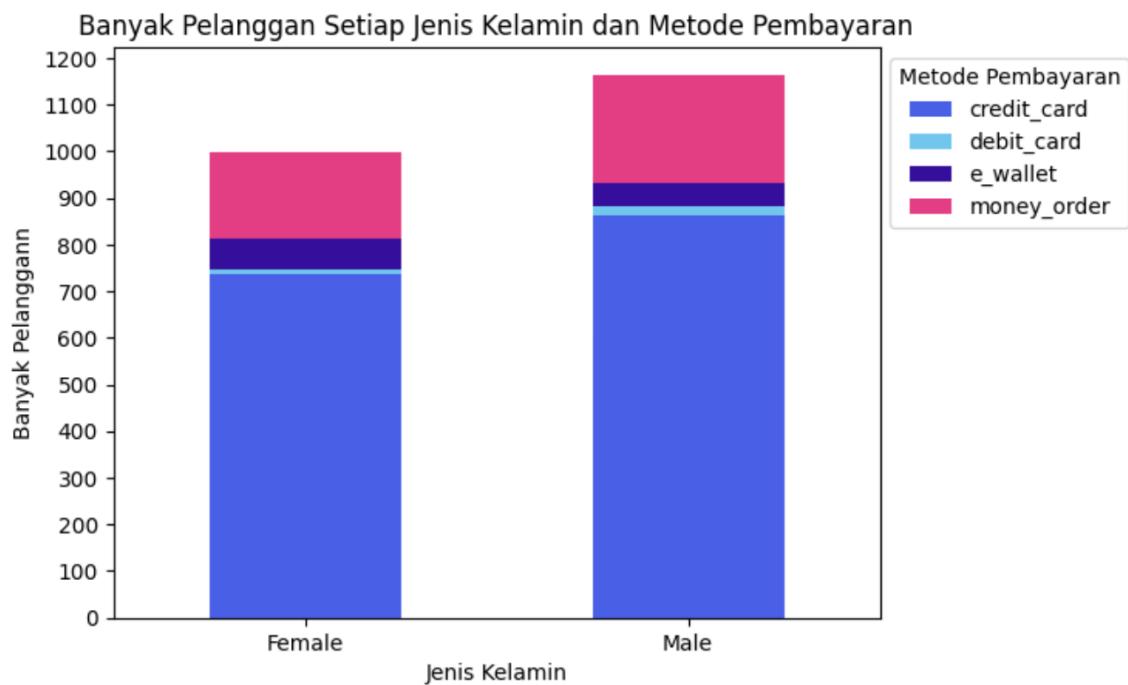
plt.figure(figsize=(12, 8))

grouped_data.plot(kind="bar", stacked=True, color= ["#4361EE", "#4CC9F0", "#3A0CA3", "#F72585"])

plt.title('Banyak Pelanggan Setiap Jenis Kelamin dan Metode Pembayaran')
plt.xlabel('Jenis Kelamin')
plt.xticks(rotation=0)
plt.ylabel('Banyak Pelanggann')
plt.yticks(range(0, int(grouped_data.max().max())+400, 100))
plt.legend(title='Metode Pembayaran', bbox_to_anchor=(1, 1))

plt.show()

```



Insight: Pelanggan lebih banyak yang berjenis kelamin laki-laki. metode pembayaran terbanyak yang dipilih kedua jenis kelamin sama, yaitu credit card, diikuti money order, e-wallet, lalu debit card.

Pemilihan warna

- Biru: menandai metode pembayaran menggunakan kartu kredit (*credit_card*)
- Pink: menandai metode pembayaran menggunakan uang tunai (*money_order*)
- Ungu: menandai metode pembayaran menggunakan uang elektronik(*e_wallet*)
- Biru Muda: menandai metode pembayaran menggunakan kartu debit (*debit_card*)

B. Penampilan Perubahan terhadap Waktu

1. Diagram Garis Frekuensi Jumlah Pesanan pada Setiap Tanggal di Bulan November

```

# Mengubah 'Order_Date' menjadi tipe data datetime
df['Order_Date'] = pd.to_datetime(df['Order_Date'], format='%m/%d/%Y')

# Menggunakan groupby untuk menghitung frekuensi pesanan untuk setiap tanggal
frekuensi = df['Order_Date'].value_counts()

# Mengurutkan indeks agar line chart lebih terorganisir
frekuensi = frekuensi.sort_index()

# Membuat line chart menggunakan seaborn
plt.figure(figsize=(16, 12))
sns.lineplot(x=frekuensi.index, y=frekuensi.values, marker='o', color='dodgerblue',
             linestyle='-', linewidth=2)

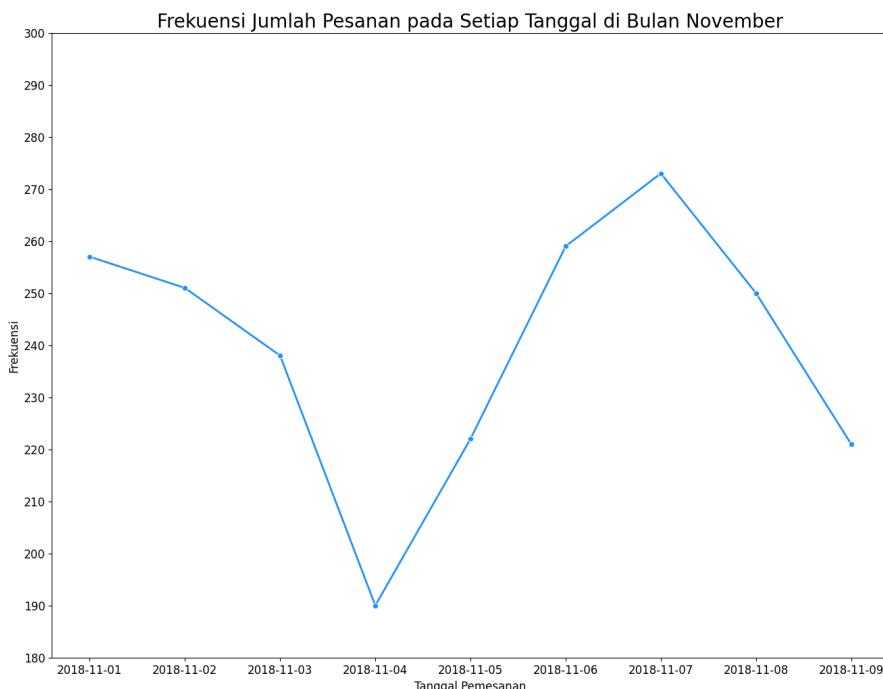
# Menambahkan label pada sumbu-sumbu
plt.title('Frekuensi Jumlah Pesanan pada Setiap Tanggal di Bulan November', fontsize=20)
plt.xlabel('Tanggal Pemesanan', fontsize=12)
plt.ylabel('Frekuensi', fontsize=12)

# Menentukan format tanggal pada sumbu x
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('20%y-%m-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator())

plt.xticks(fontsize=12)
plt.yticks(range(180, max(frekuensi.values)+ 30, 10), fontsize=12)

# Menampilkan plot
plt.show()

```



Insight: Jumlah pemesanan terbanyak terjadi pada tanggal 7 November.

Pemilihan warna: warna biru dipilih untuk memperjelas hasil dari diagram garis.

2. Diagram Batang Jumlah Kuantitas Pembelian berdasarkan Tanggal Pemesanan

```
# Mengubah 'Order_Date' menjadi tipe data datetime dengan format yang sesuai
df['Order_Date'] = pd.to_datetime(df['Order_Date'], format='%m/%d/%Y')

# Menggunakan groupby untuk menghitung total kuantitas pembelian untuk setiap tanggal
total_quantity = df.groupby('Order_Date')['Quantity'].sum().reset_index()

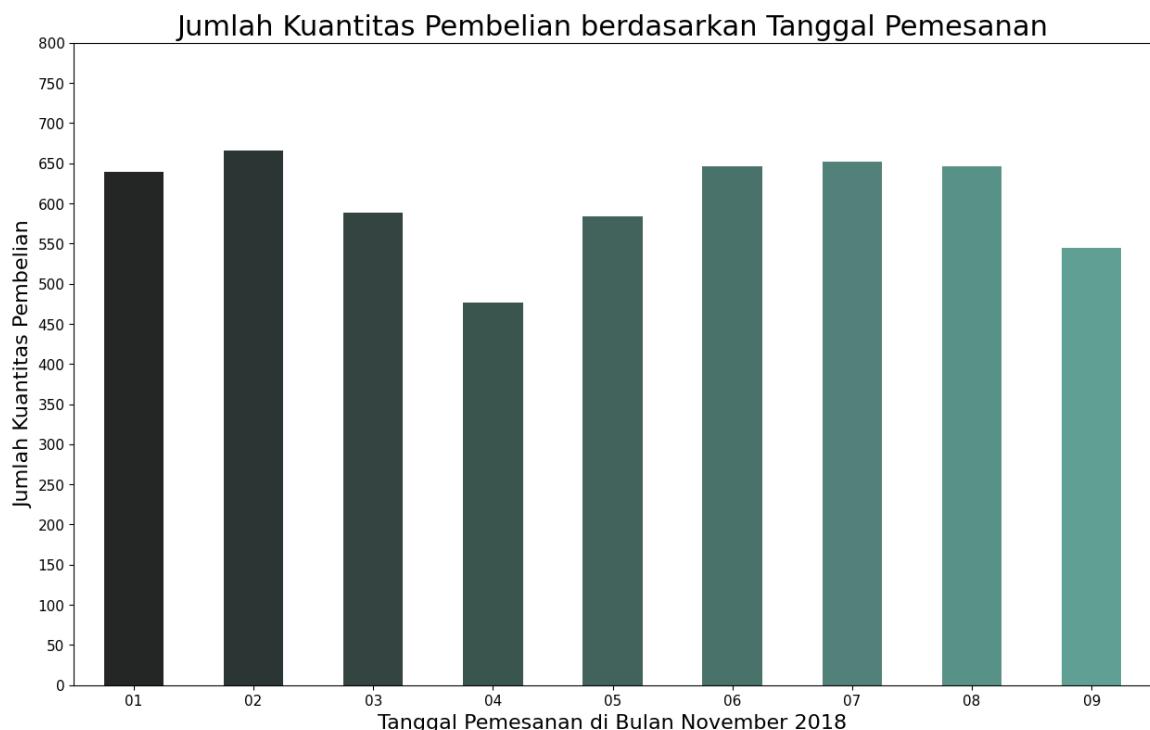
plt.figure(figsize=(15, 9))
sns.barplot(data=total_quantity, x='Order_Date', y='Quantity', palette="dark:#5A9", width=0.5)

plt.title('Jumlah Kuantitas Pembelian berdasarkan Tanggal Pemesanan', fontsize=22)
plt.xlabel('Tanggal Pemesanan di Bulan November 2018', fontsize=16)
plt.ylabel('Jumlah Kuantitas Pembelian', fontsize=16)

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator())

plt.xticks(rotation=0, fontsize=11)
plt.yticks(range(0, max(total_quantity['Quantity'])+150, 50), fontsize=11)

plt.show()
```



Insight: Jumlah kuantitas pembelian terbanyak terjadi pada tanggal 2 November.

Pemilihan warna:

Dilakukan gradasi warna dengan tema '`dark:#5A9`' untuk membedakan hasil jumlah kuantitas pembelian dari setiap tanggal di bulan November pada diagram batang.

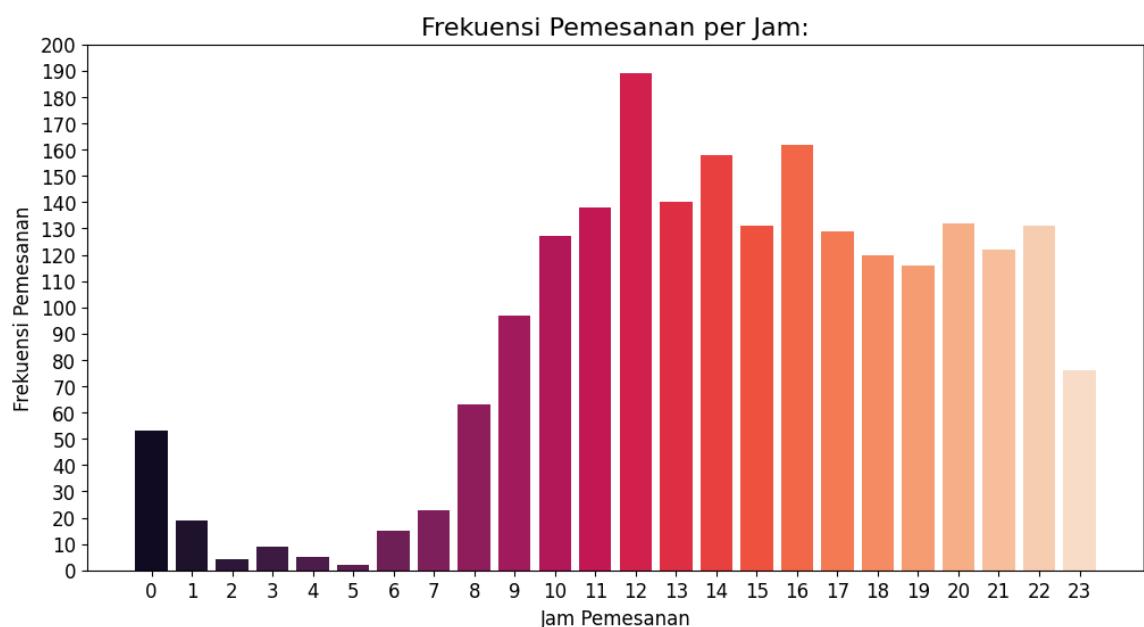
3. Diagram Batang Frekuensi Pemesanan per Jam

```
# Mengubah nilai 'Time' ke bentuk jam
df['Hour'] = pd.to_datetime(df['Time']).dt.hour
total_transactions = df.groupby('Hour').size()

sns.set_palette("rocket", n_colors=len(total_transactions))

plt.figure(figsize=(12, 6))
for i, (index, value) in enumerate(total_transactions.items()):
    plt.bar(index, value, color=sns.color_palette()[i])

plt.title('Frekuensi Pemesanan per Jam:', fontsize=16)
plt.xlabel('Jam Pemesanan', fontsize=12)
plt.xticks(fontsize=12, ticks=range(0, 24, 1))
plt.ylabel('Frekuensi Pemesanan', fontsize=12)
plt.yticks(range(0, max(total_transactions.values) + 20, 10), fontsize=12)
plt.show()
```



Insight: Frekuensi pemesanan tertinggi terjadi pada jam 12.

Pemilihan warna:

Dilakukan gradasi warna dengan tema '*rocket*' untuk membedakan hasil frekuensi pemesanan dari setiap jam pada diagram batang.

4. Histogram Frekuensi Pemesanan per Jam berdasarkan Prioritas Order

```
# Mengubah nilai 'Time' ke bentuk jam
df['Hour'] = pd.to_datetime(df['Time']).dt.hour

palette = sns.color_palette(['darkgreen', 'gold', 'darkorange', 'crimson'])

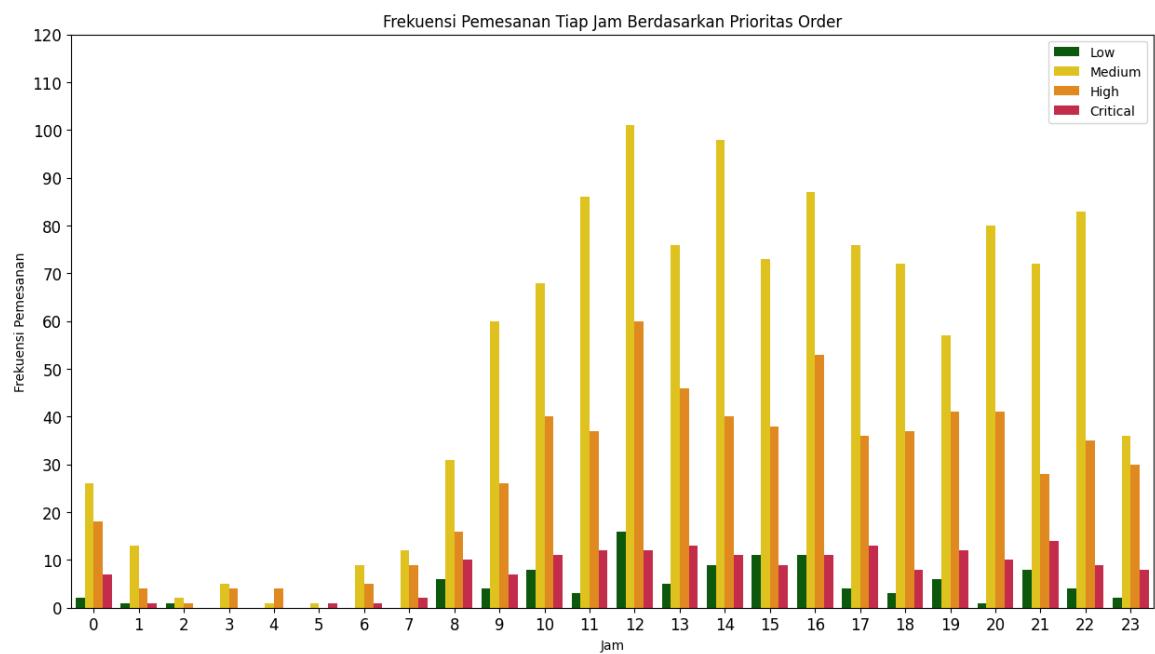
df_plot = df.groupby(['Hour', 'Order_Priority']).size().reset_index(name='Total Transactions')

plt.figure(figsize=(15, 8))

sns.barplot(x='Hour', y='Total Transactions', hue='Order_Priority', data=df_plot,
            palette=palette, order=range(24), hue_order=['Low', 'Medium', 'High', 'Critical'], alpha=1.0)

plt.xlabel('Jam')
plt.xticks(fontsize=12, ticks=range(0, 24, 1))
plt.ylabel('Frekuensi Pemesanan')
plt.yticks(range(0, df_plot['Total Transactions'].max() + 20, 10), fontsize=12)
plt.title('Frekuensi Pemesanan Tiap Jam Berdasarkan Prioritas Order')
plt.legend()

plt.show()
```



Insight: Berdasarkan semua prioritas order, baik low, medium, high, maupun critical frekuensi pemesanan tertinggi terjadi pada jam 12.

Pemilihan warna:

- Merah: menandai prioritas critical (sangat tinggi)
- Oranye: menandai prioritas high (tinggi)
- Kuning: menandai prioritas medium (menengah)
- Hijau: menandai prioritas low (rendah)

C. Penampilan Hierarki dan Hubungan Keseluruhan-Bagian

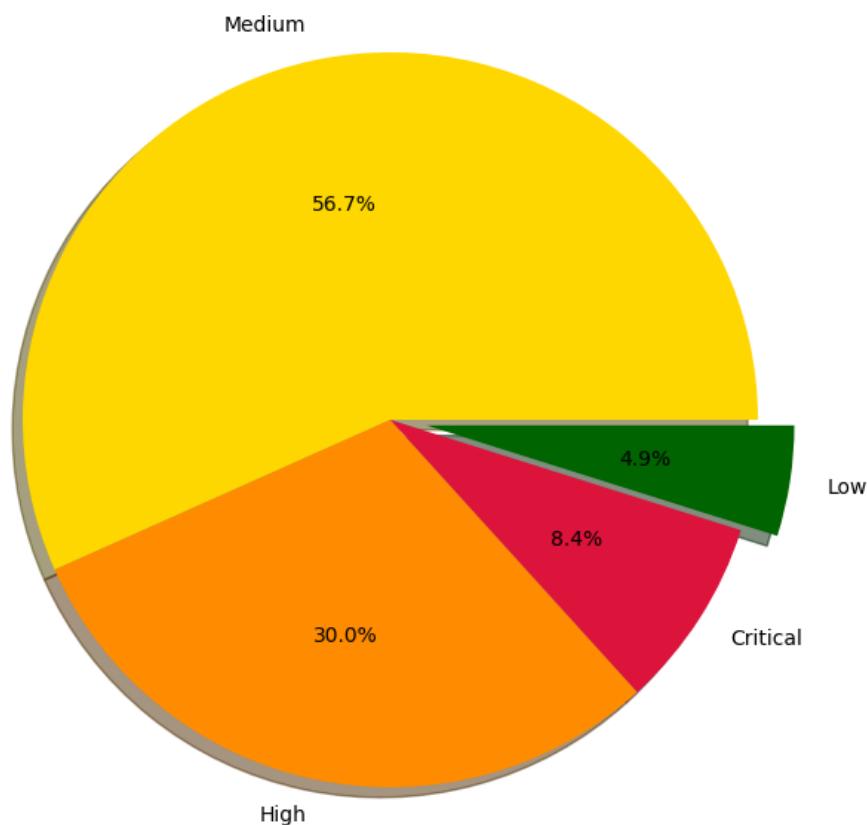
1. Diagram Pie Perbandingan Jumlah Pelanggan berdasarkan Prioritas Pesanan.

```
Jumlah = df['Order_Priority'].value_counts()

plt.figure(figsize=(8, 10))

colors = sns.color_palette(['gold', 'darkorange', 'crimson', 'darkgreen'])
Jumlah.plot(kind="pie", title="Perbandingan Jumlah Pelanggan berdasarkan Prioritas Pesanan",
            autopct='%.1f%%', colors=colors, explode=(0, 0, 0, 0.1), shadow=True)
plt.ylabel('') # Menghilangkan label sumbu y
plt.show()
```

Perbandingan Jumlah Pelanggan berdasarkan Prioritas Pesanan



Insight: Urutan jumlah pelanggan terbanyak berdasarkan prioritas pesanan adalah medium, high, critical, dan low.

Pemilihan warna:

- Merah: menandai prioritas critical (sangat tinggi)
- Oranye: menandai prioritas high (tinggi)
- Kuning: menandai prioritas medium (menengah)
- Hijau: menandai prioritas low (rendah)

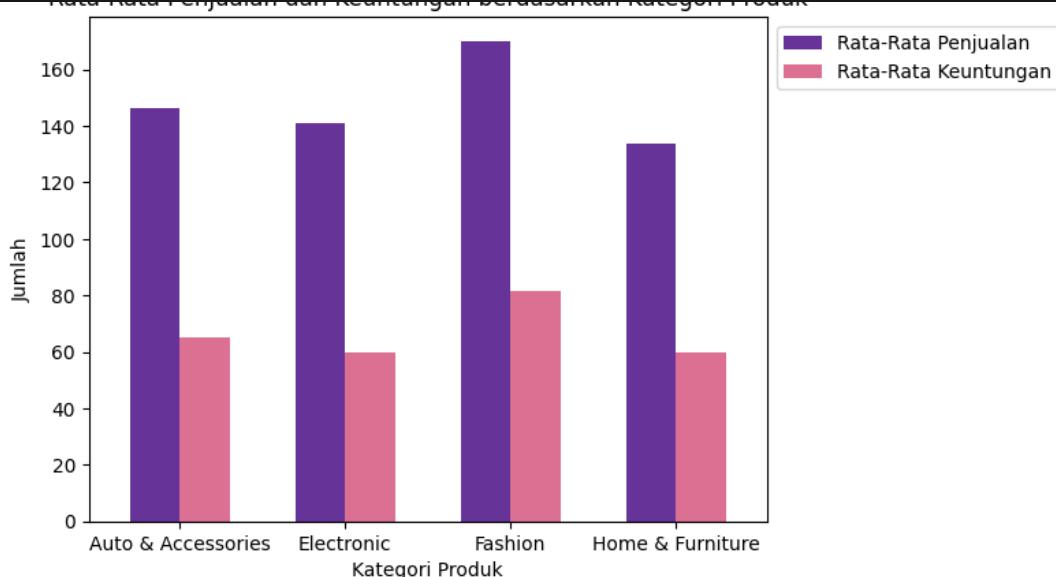
2. Diagram Bar Rata-Rata Penjualan dan Keuntungan berdasarkan Kategori Produk

```
average_data = df.groupby("Product_Category")[["Sales", "Profit"]].mean()

plt.figure(figsize=(12, 8))
average_data.plot(kind="bar", width=0.6, color=['rebeccapurple', 'palevioletred'])

plt.title('Rata-Rata Penjualan dan Keuntungan berdasarkan Kategori Produk')
plt.xlabel('Kategori Produk')
plt.xticks(rotation=0)
plt.ylabel('Jumlah')

plt.legend(["Rata-Rata Penjualan", "Rata-Rata Keuntungan"], bbox_to_anchor=(1, 1))
plt.show()
```



Insight: Kategori Fashion memiliki rata-rata tertinggi dari penjualan produk dan keuntungan yang didapatkan dari diskon. Semakin tinggi penjualan dari produk yang dibeli pelanggan, semakin besar keuntungan yang didapatkan dari pelanggan. Urutan rata-rata penjualan produk dan keuntungan pelanggan dari tertinggi ke terendah adalah Fashion, Auto & Accessories, Electronic, dan Home & Furniture.

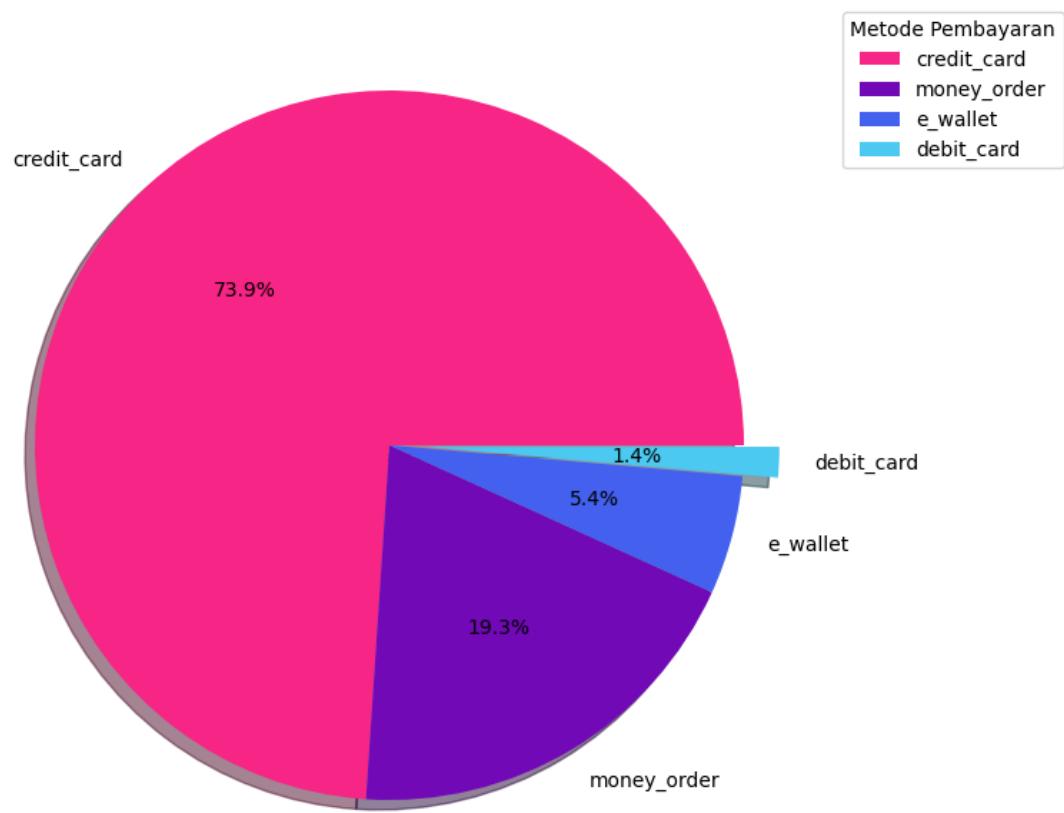
Pemilihan warna:

- Ungu: menandai rata - rata penjualan (*sales*)
- Pink: menandai rata - rata keuntungan (*profit*)

3. Diagram Pie Perbandingan Jumlah Pelanggan berdasarkan Metode Pembayaran

```
Jumlah = df['Payment_method'].value_counts()
plt.figure(figsize=(8, 10))
Jumlah.plot(kind="pie", title="Perbandingan Jumlah Pelanggan berdasarkan Metode Pembayaran",
            autopct='%.1f%%', colors=["#F72585", "#7209B7", "#4361EE", "#4CC9F0"], explode=(0, 0, 0, 0.1), shadow=True)
plt.ylabel('')
plt.legend(title='Metode Pembayaran', bbox_to_anchor=(1, 1))
plt.show()
```

Perbandingan Jumlah Pelanggan berdasarkan Metode Pembayaran



Insight: Urutan metode pembayaran dengan jumlah pelanggan terbanyak dari tertinggi ke terendah adalah credit card, money order, e-wallet, dan debit card.

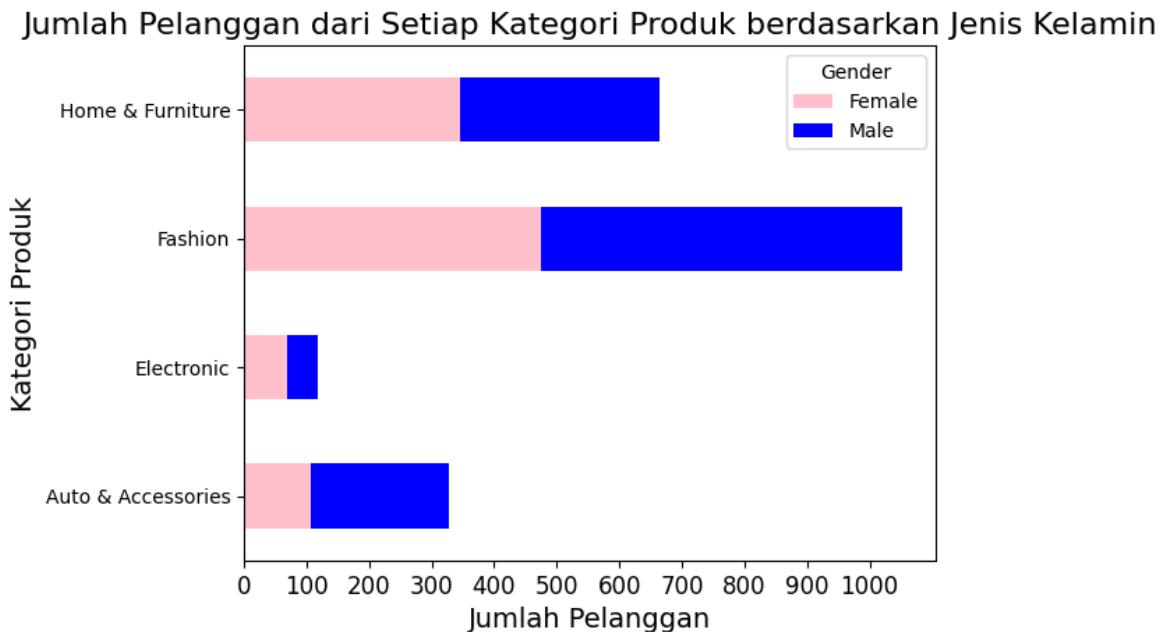
Pemilihan warna:

- Pink: menandai metode pembayaran menggunakan kartu kredit (*credit_card*)
- Ungu: menandai metode pembayaran menggunakan uang tunai (*money_order*)
- Biru: menandai metode pembayaran menggunakan uang elektronik(*e_money*)
- Biru Muda: menandai metode pembayaran menggunakan kartu debit (*debit_card*)

4. Diagram Bar Jumlah Pelanggan dari Setiap Kategori berdasarkan Jenis Kelamin

```
count_df = df.groupby(['Product_Category', 'Gender']).size().unstack()
plt.figure(figsize=(10, 6))
ax = count_df.plot(kind='barh', stacked=True, color=['pink', 'blue'])

plt.title('Jumlah Pelanggan dari Setiap Kategori Produk berdasarkan Jenis Kelamin', fontsize=16)
plt.xlabel('Jumlah Pelanggan', fontsize=14)
plt.xticks(range(0, count_df.max().max() + 500, 100), fontsize=12)
plt.ylabel('Kategori Produk', fontsize=14)
plt.legend(title='Gender')
plt.show()
```



Insight: Pelanggan laki-laki lebih banyak membeli produk dengan kategori Fashion dan Auto & Accessories. Sedangkan, pelanggan perempuan lebih banyak membeli produk dengan kategori Home & Furniture dan Electronic. Urutan kategori produk dengan jumlah pelanggan terbanyak dari tertinggi ke terendah adalah Fashion, Home & Furniture, Auto & Accessories, dan Electronic.

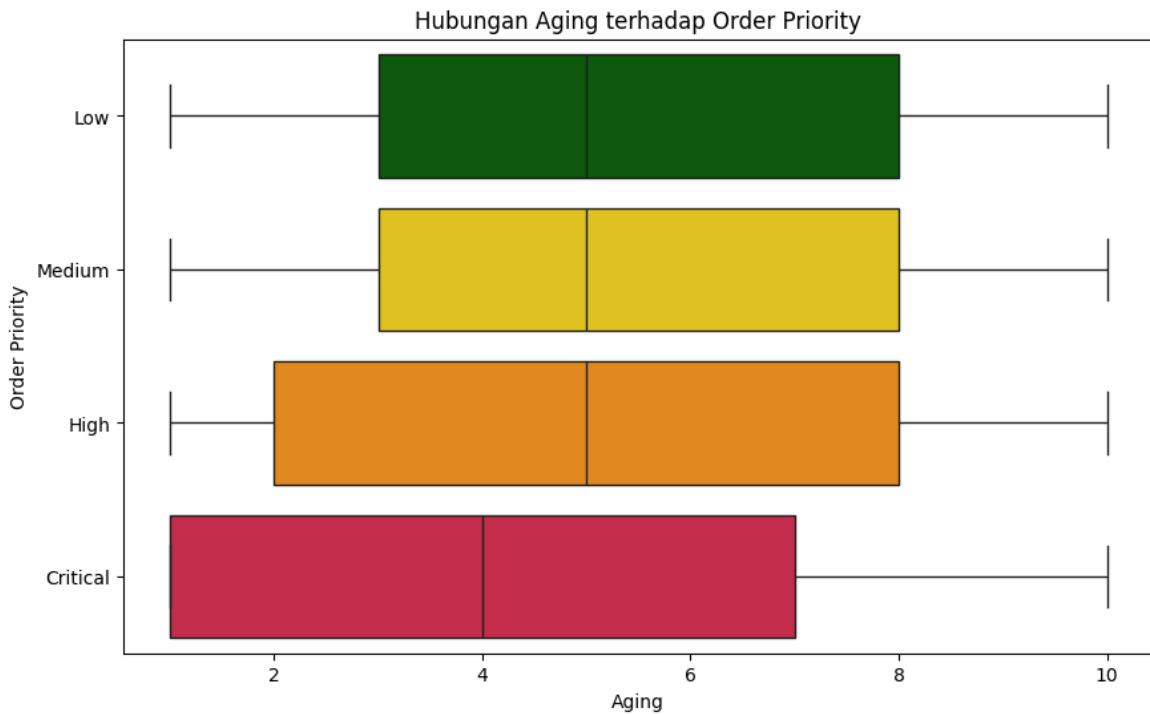
Pemilihan warna:

- Pink: menandai jenis kelamin pelanggan perempuan (*female*)
- Biru: menandai jenis kelamin pelanggan laki - laki (*male*)

D. Plotting Relationships

1. Box Plot Hubungan Aging terhadap Order Priority

```
plt.figure(figsize=(10, 6))
palette = sns.color_palette(['darkgreen', 'gold', 'darkorange', 'crimson'])
sns.boxplot(x='Aging', y='Order_Priority', data=df, order=['Low', 'Medium', 'High', 'Critical'], palette=palette)
plt.title('Hubungan Aging terhadap Order Priority')
plt.xlabel('Aging')
plt.ylabel('Order Priority')
plt.show()
```



Insight: Order priority critical memiliki waktu pengiriman terendah, dengan range dari 1-7 hari dan median/titik tengah 4 hari. Order priority lainnya memiliki median waktu pengiriman yang sama, yaitu 5 hari. Namun, dibanding dengan order priority Low dan Medium, order Priority High memiliki range lebih besar, dari 2-8 hari. Order priority Low dan Medium sama saja pada range dan mediannya.

Pemilihan warna:

- Merah: menandai prioritas critical (sangat tinggi)
- Oranye: menandai prioritas high (tinggi)
- Kuning: menandai prioritas medium (menengah)
- Hijau: menandai prioritas low (rendah)

2. Scatter Plot Hubungan antara Keuntungan terhadap Biaya Pengiriman

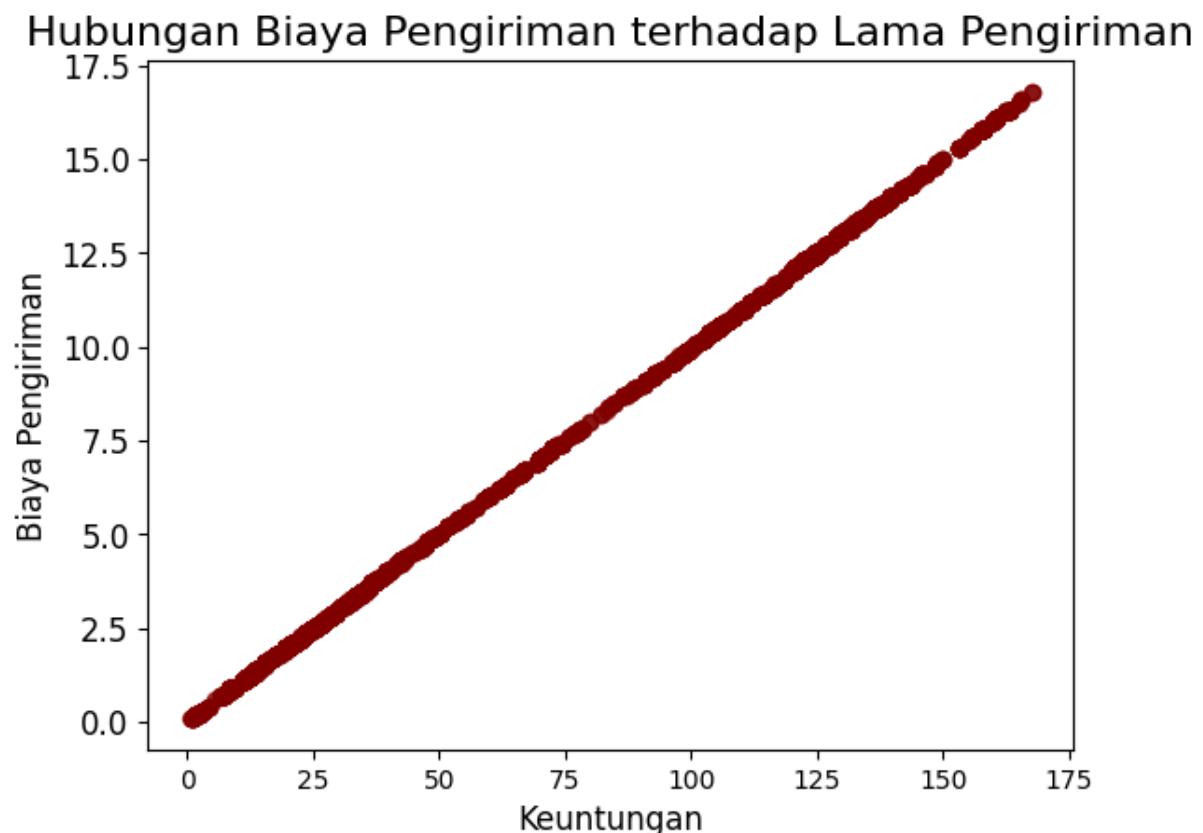
```
plt.scatter(df['Profit'], df['Shipping_Cost'], alpha=0.7, color='maroon')
plt.title('Hubungan Biaya Pengiriman terhadap Keuntungan', fontsize=16)
plt.xlabel('Keuntungan', fontsize=12)
plt.ylabel('Biaya Pengiriman', fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```

Insight: Harga biaya pengiriman bersifat linear dengan lama pengiriman produk. Berdasarkan grafik, biaya pengiriman adalah sekitar 10% dari keuntungan yang diperoleh pelanggan.

Pemilihan warna:

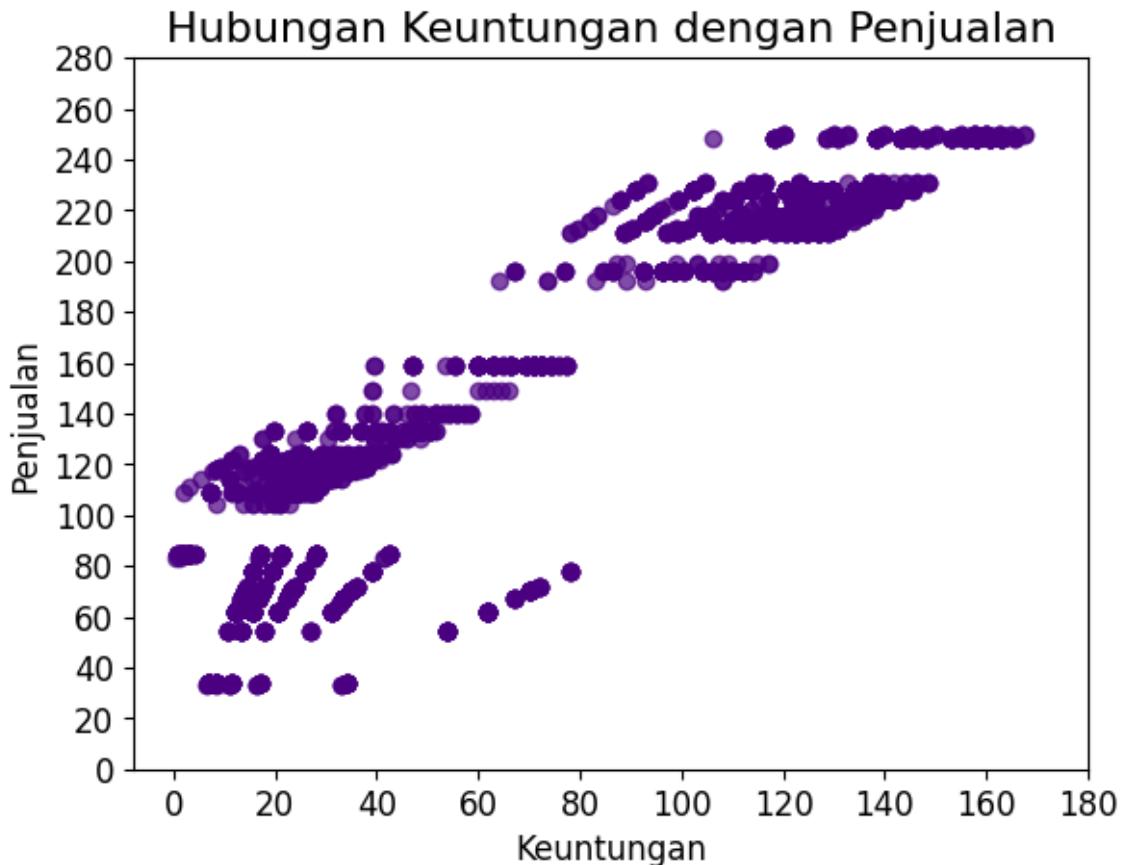
Merah: menandakan hasil *plotting* hubungan antara Biaya Pengiriman (*shipping cost*) terhadap keuntungan (*profit*)

3. Scatter Plot Hubungan Keuntungan dengan Penjualan



```
plt.scatter(df['Profit'], df['Sales'], alpha=0.7, color='indigo')
plt.xticks(range(0, int(df['Profit'].max()) + 20, 20), fontsize=12)
plt.yticks(range(0, int(df['Sales'].max()) + 40, 20), fontsize=12)
plt.title('Hubungan Keuntungan dengan Penjualan', fontsize=16)
plt.xlabel('Keuntungan', fontsize=12)
plt.ylabel('Penjualan', fontsize=12)

plt.show()
```



Insight: semakin besar keuntungan yang didapatkan oleh pelanggan, semakin tinggi penjualan produk yang dibeli. Maka, produk dengan penjualan yang lebih tinggi cenderung memberikan keuntungan lebih besar untuk pelanggan.

Pemilihan warna:

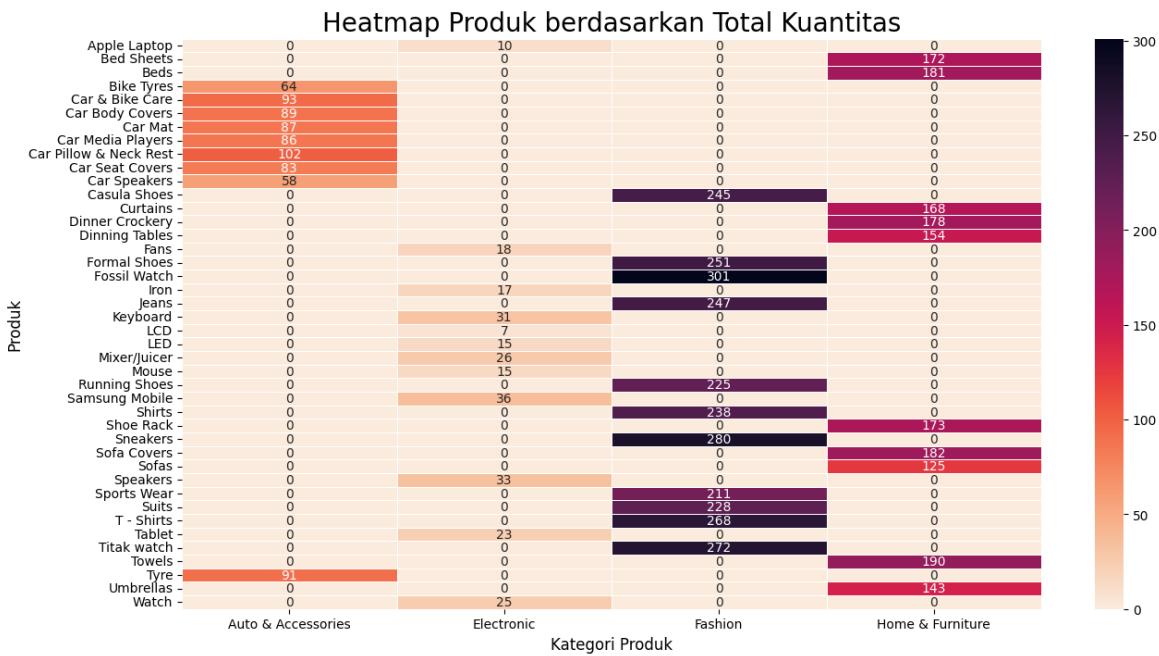
Ungu: menandakan hasil *plotting* hubungan antara penjualan (*sales*) terhadap keuntungan (*profit*)

4. Heatmap Produk berdasarkan Total Kuantitas

```
total_quantity = df.groupby(['Product_Category', 'Product'])['Quantity'].sum().reset_index()
total_quantity_sorted = total_quantity.sort_values(by='Quantity', ascending=False)

plt.figure(figsize=(15, 8))
heatmap_data = total_quantity_sorted.pivot_table(index='Product', columns='Product_Category', values='Quantity', aggfunc='sum', fill_value=0)
sns.heatmap(heatmap_data, cmap='rocket_r', annot=True, fmt='g', linewidths=.5)

plt.xlabel('Kategori Produk', fontsize=12)
plt.ylabel('Produk', fontsize=12)
plt.title('Heatmap Produk berdasarkan Total Kuantitas', fontsize=20)
plt.show()
```



Insight: Kategori produk dengan kuantitas terbanyak adalah Fashion, diikuti dengan Home Furniture, Auto & Accessories, dan Electronic

Pemilihan warna:

Semakin gelap warna yang diberikan pada tabel, maka semakin besar kuantitas pembelian produk dari setiap produk kategori.

5. Bubble Plot Biaya Pengiriman terhadap Penjualan berdasarkan Kategori Produk

```

Ongkir = df['Shipping_Cost']
Penjualan = df['Sales']
product_category = df['Product_Category']

# Mapping nilai product category ke warna
color_mapping = {'Auto & Accessories': 'green', 'Fashion': 'yellow', 'Home & Furniture': 'orange', 'Electronic': 'red'}
colors = product_category.map(color_mapping)

# Menentukan ukuran bubble (variabel ketiga)
bubble_size = product_category.map({'Auto & Accessories': 40, 'Fashion': 80, 'Home & Furniture': 65, 'Electronic': 100})

# Membuat bubble chart
plt.figure(figsize=(12, 8))
plt.scatter(Ongkir, Penjualan, s=bubble_size, alpha=0.7, c=colors.values, edgecolors='w', linewidth=1)

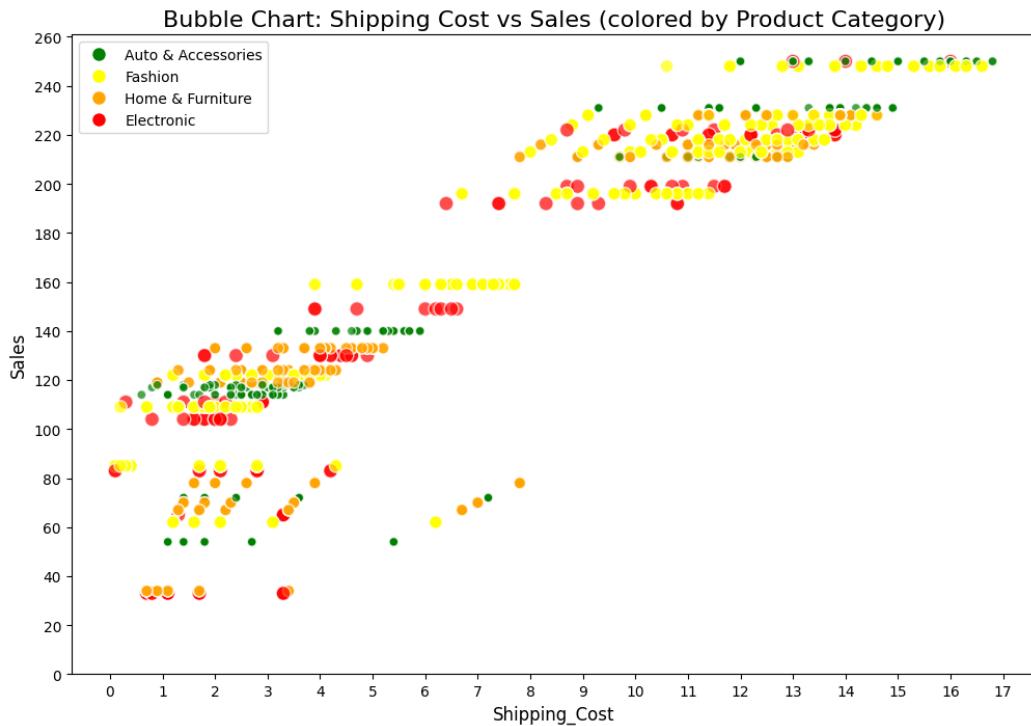
# Menambahkan label pada sumbu-sumbu
plt.xlabel('Shipping_Cost', fontsize=12)
plt.ylabel('Sales', fontsize=12)
plt.title('Bubble Chart: Shipping Cost vs Sales (colored by Product Category)', fontsize=16)

# Menetapkan interval sumbu x dan y
plt.xticks(np.arange(0, Ongkir.max() + 0.5, 1), fontsize=10)
plt.yticks(np.arange(0, Penjualan.max() + 20, 20), fontsize=10)

# Menampilkan legend untuk warna product category
legend_labels = [plt.Line2D([0], [0], marker='o', color='w', label=label, markerfacecolor=color, markersize=10) for label, color in color_mapping.items()]
plt.legend(handles=legend_labels)

# Menampilkan plot
plt.show()

```



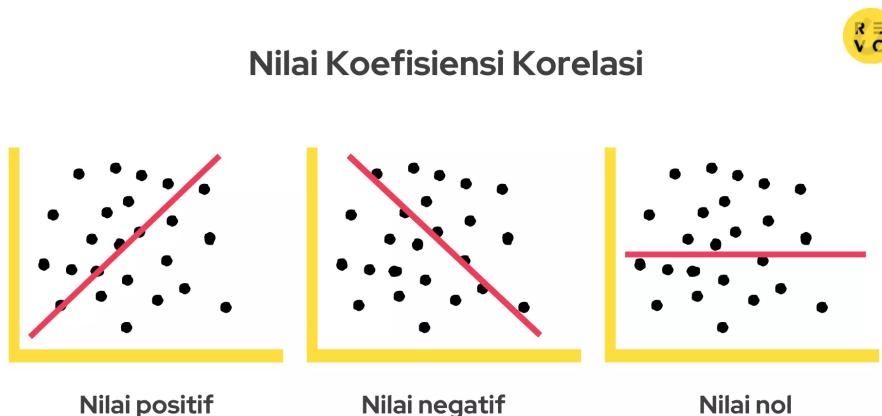
Insight: Produk dengan sales yang lebih banyak akan membutuhkan biaya pengiriman yang lebih besar, berlaku untuk setiap jenis kategori produk.

Pemilihan warna:

- Merah: menandai produk dengan kategori elektronik (*Electronic*)
- Oranye: menandai produk dengan kategori perabot rumah tangga (*Home & Furniture*)
- Kuning: menandai produk dengan kategori mode (*Fashion*)
- Hijau: menandai produk dengan kategori otomotif dan aksesorisnya (*Auto & Accessories*)

3.4. Korelasi

Analisis data menggunakan korelasi ditujukan untuk menunjukkan adanya hubungan antara dua variabel. Analisis korelasi menghitung nilai koefisien korelasi antara kedua variabel dengan range output $-1 < r < 1$.



Gambar 2.4.1 Ilustrasi Nilai Koefisien Korelasi pada Grafik. Sumber: Investopedia

Berdasarkan nilai koefisien korelasi yang didapatkan, ada 3 kemungkinan yang dapat terjadi. Kemungkinan pertama adalah terdapat korelasi lurus/positif antara kedua variabel apabila nilai koefisien korelasinya mendekati 1. Sebaliknya, korelasi terbalik/negatif terjadi apabila nilai koefisien korelasinya mendekati -1. Apabila koefisien korelasi mendekati 0, maka tidak ada korelasi antara kedua variabel.

Cara menyimpulkan jenis korelasi antara kedua variabel pada suatu grafik dilakukan dengan melihat bentuk garis nilai korelasinya. Semakin miring ke atas garisnya, semakin positif korelasinya. Semakin lurus garisnya, semakin tidak ada hubungan antara kedua variabel. Semakin miring ke bawah garisnya, semakin negatif korelasinya.

Korelasi didapatkan dari:

```
import pandas as pd  
import matplotlib.pyplot as plt  
df = pd.read_csv("/content/E-Com Dataset.csv")
```

Berikut hasil yang didapat dari korelasi data

A. Aging

1. Korelasi antara Aging dengan Shipping Cost

```

from matplotlib.lines import Line2D

correlation = df['Aging'].corr(df['Shipping_Cost'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Aging', y='Shipping_Cost', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

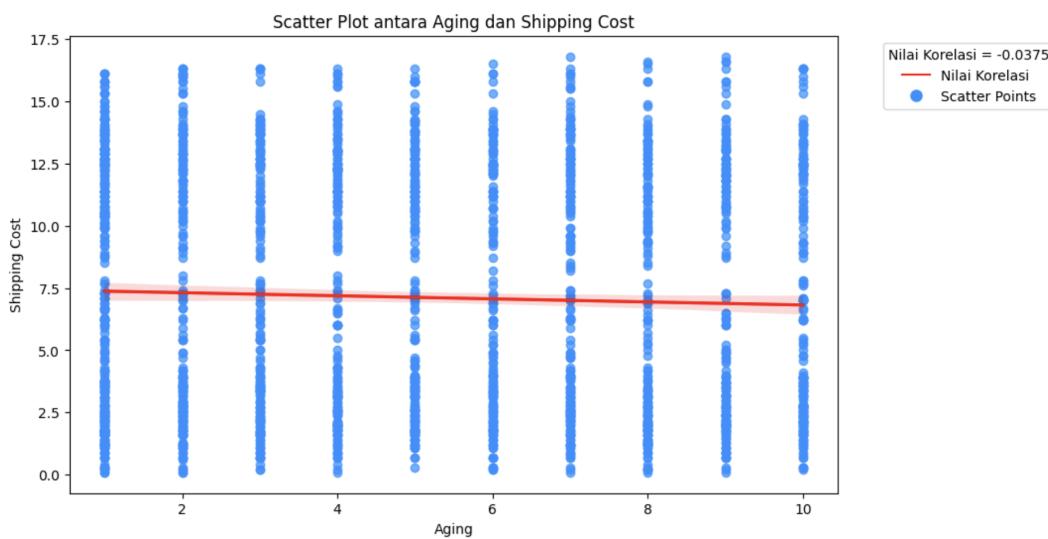
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Aging dan Shipping Cost')
plt.xlabel('Aging')
plt.ylabel('Shipping Cost')

plt.show()

```

Nilai korelasinya adalah -0.03753499929509522
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena koefisien korelasinya mendekati 0.

2. Korelasi antara Aging dengan Profit

```

from matplotlib.lines import Line2D

correlation = df['Aging'].corr(df['Profit'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Aging', y='Profit', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]

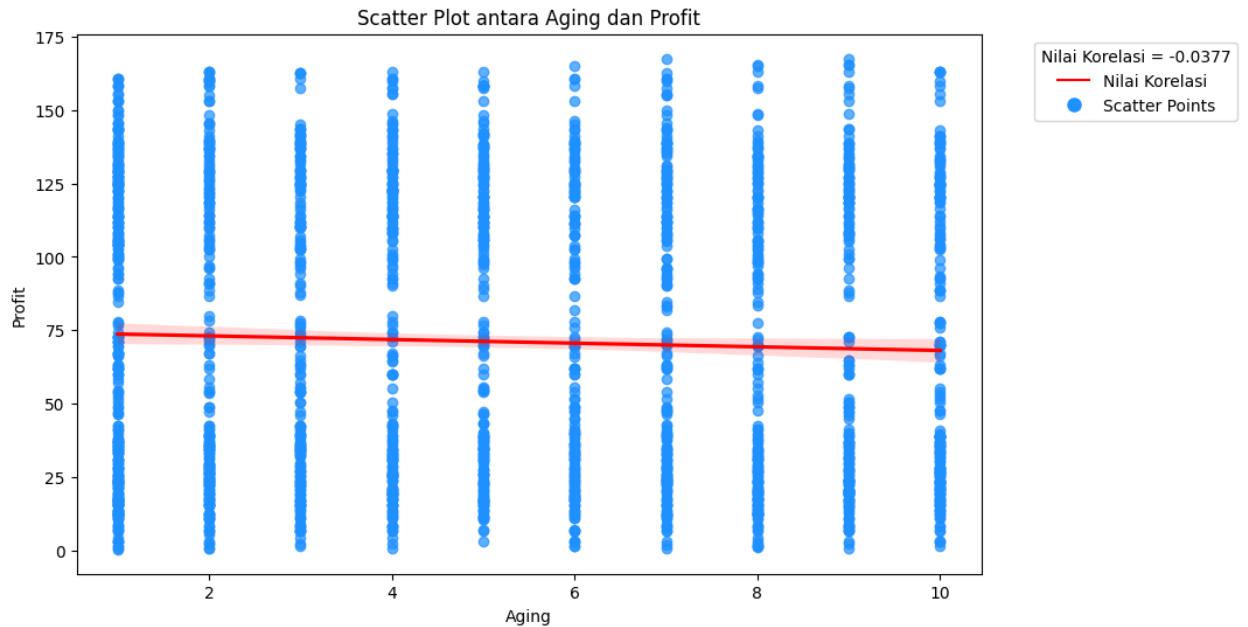
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Aging dan Profit')
plt.xlabel('Aging')
plt.ylabel('Profit')

plt.show()

```

Nilai korelasinya adalah -0.037719141294457664
 Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena koefisien korelasinya mendekati 0.

3. Korelasi antara Aging dengan Sales

```

from matplotlib.lines import Line2D

correlation = df['Aging'].corr(df['Sales'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Aging', y='Sales', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

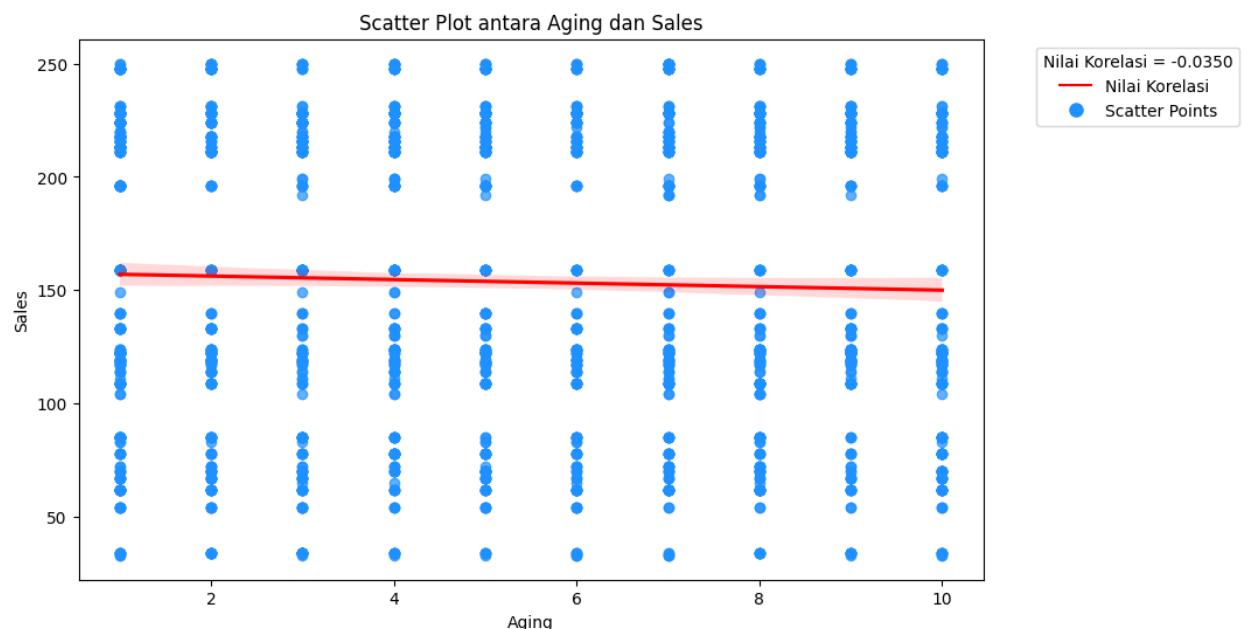
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')
]
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Aging dan Sales')
plt.xlabel('Aging')
plt.ylabel('Sales')

plt.show()

```

Nilai korelasinya adalah -0.03496045938464408
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena koefisien korelasinya mendekati 0.

4. Korelasi antara Aging dengan Discount

```

from matplotlib.lines import Line2D

correlation = df['Aging'].corr(df['Discount'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Aging', y='Discount', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
]

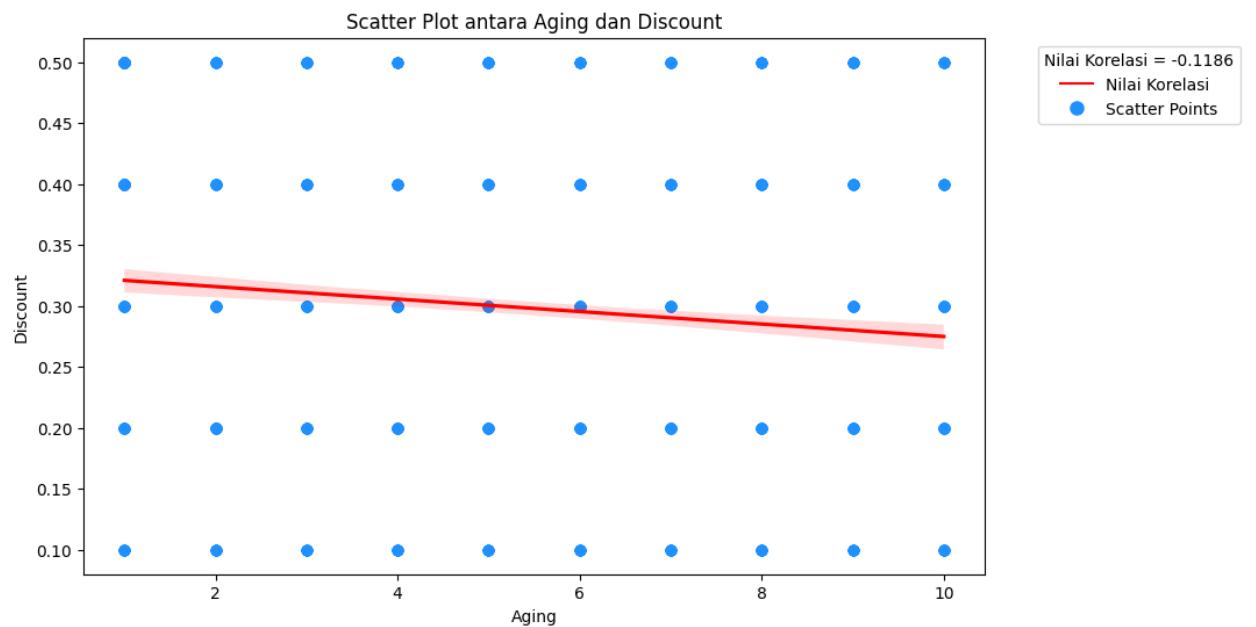
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Aging dan Discount')
plt.xlabel('Aging')
plt.ylabel('Discount')

plt.show()

```

Nilai korelasinya adalah -0.11857193323844717
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena koefisien korelasinya mendekati 0.

5. Korelasi antara Aging dengan Quantity

```

from matplotlib.lines import Line2D

correlation = df['Aging'].corr(df['Quantity'])
print(f'Nilai korelasinya adalah {correlation}')

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.replot(x='Aging', y='Quantity', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')
]

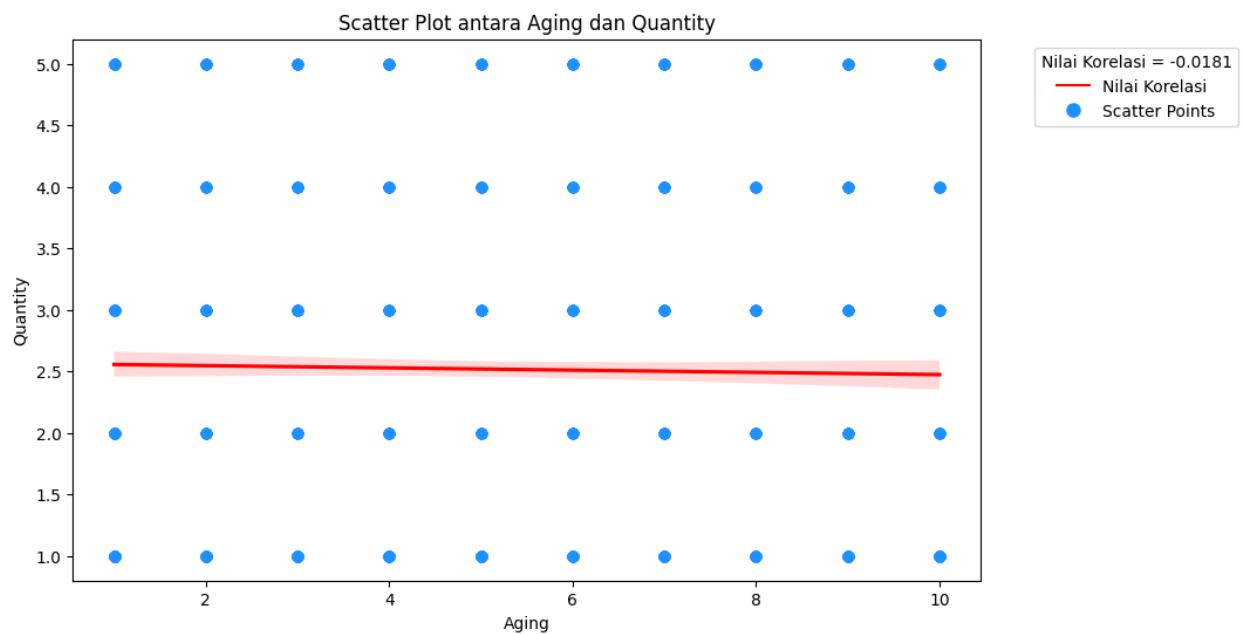
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Aging dan Quantity')
plt.xlabel('Aging')
plt.ylabel('Quantity')

plt.show()

```

Nilai korelasinya adalah -0.01809719299097352
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena koefisien korelasinya mendekati 0.

B. Shipping Cost

1. Korelasi antara Shipping Cost dengan Aging

```

from matplotlib.lines import Line2D

correlation = df['Shipping_Cost'].corr(df['Aging'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Shipping_Cost', y='Aging', data=df, color='red',
            scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')
]

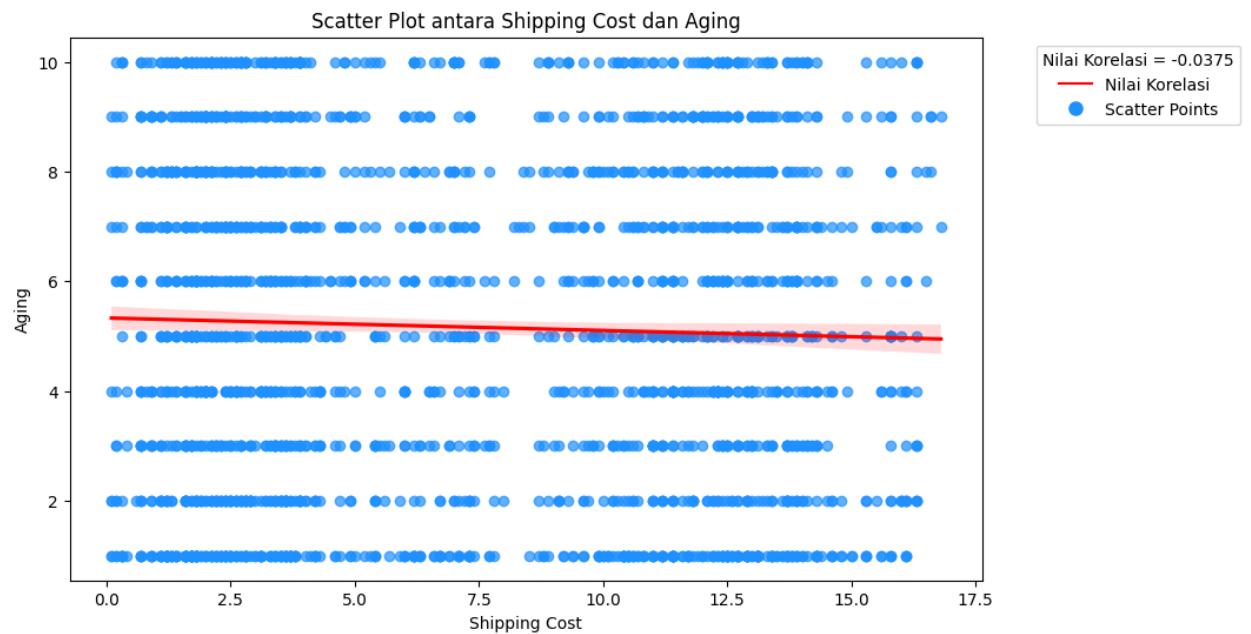
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Shipping Cost dan Aging')
plt.xlabel('Shipping Cost')
plt.ylabel('Aging')

plt.show()

```

Nilai korelasinya adalah -0.03753499929509521
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena koefisien korelasinya mendekati 0.

2. Korelasi antara Shipping Cost dengan Profit

```

from matplotlib.lines import Line2D

correlation = df['Shipping_Cost'].corr(df['Profit'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Shipping_Cost', y='Profit', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Shipping Cost dan Profit')
plt.xlabel('Shipping Cost')
plt.ylabel('Profit')

plt.show()

```

Nilai korelasinya adalah 0.9999814528852783
 Kedua nilai berkorelasi lurus



Kedua atribut berkorelasi karena memiliki koefisien korelasi mendekati 1.

3. Korelasi antara Shipping Cost dengan Discount

```

from matplotlib.lines import Line2D

correlation = df['Shipping_Cost'].corr(df['Discount'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Shipping_Cost', y='Discount', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]

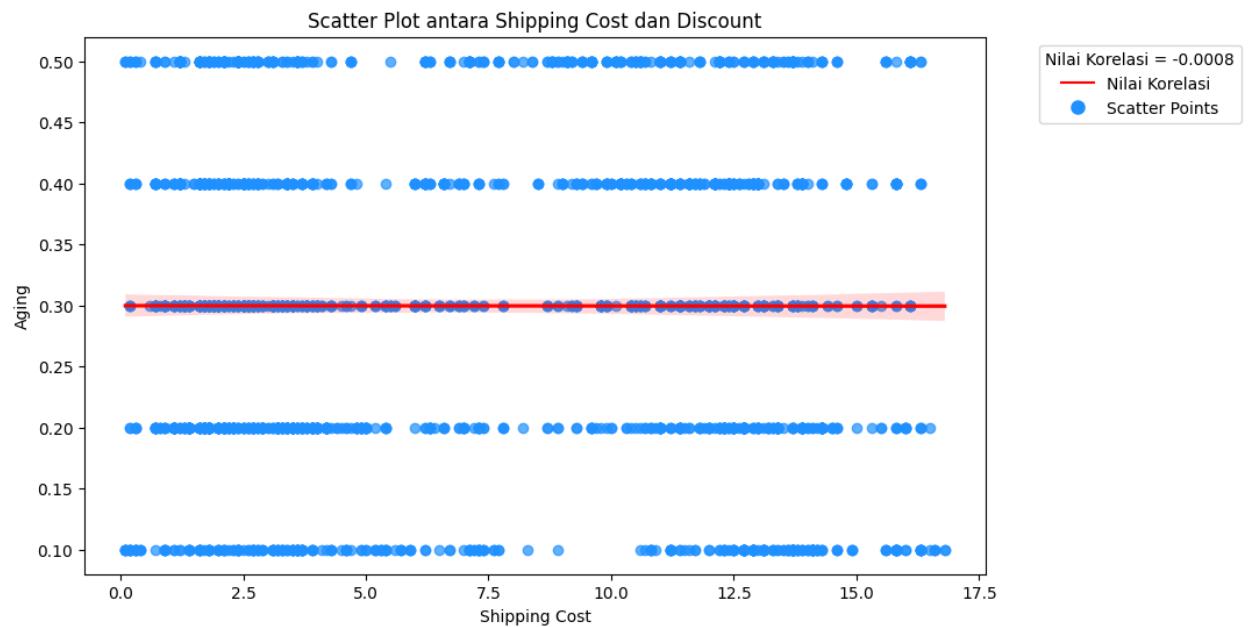
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Shipping Cost dan Discount')
plt.xlabel('Shipping Cost')
plt.ylabel('Aging')

plt.show()

```

Nilai korelasinya adalah -0.0007903480762710723
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena koefisien korelasinya mendekati 0.

4. Korelasi antara Shipping Cost dengan Sales

```

from matplotlib.lines import Line2D

correlation = df['Shipping_Cost'].corr(df['Sales'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Shipping_Cost', y='Sales', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

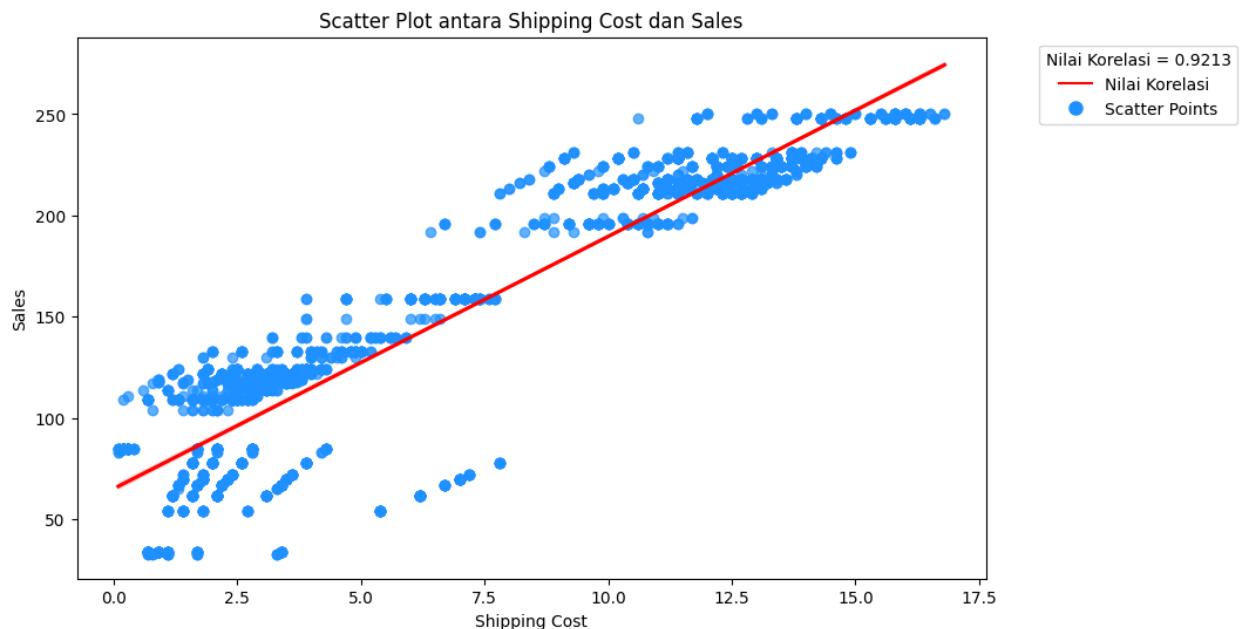
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
    
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Shipping Cost dan Sales')
plt.xlabel('Shipping Cost')
plt.ylabel('Sales')

plt.show()

```

Nilai korelasinya adalah 0.9213054279092487
Kedua nilai berkorelasi lurus



Kedua atribut berkorelasi karena memiliki koefisien korelasi mendekati 1.

5. Korelasi antara Shipping Cost dengan Quantity

```

from matplotlib.lines import Line2D

correlation = df['Shipping_Cost'].corr(df['Quantity'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Shipping_Cost', y='Quantity', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

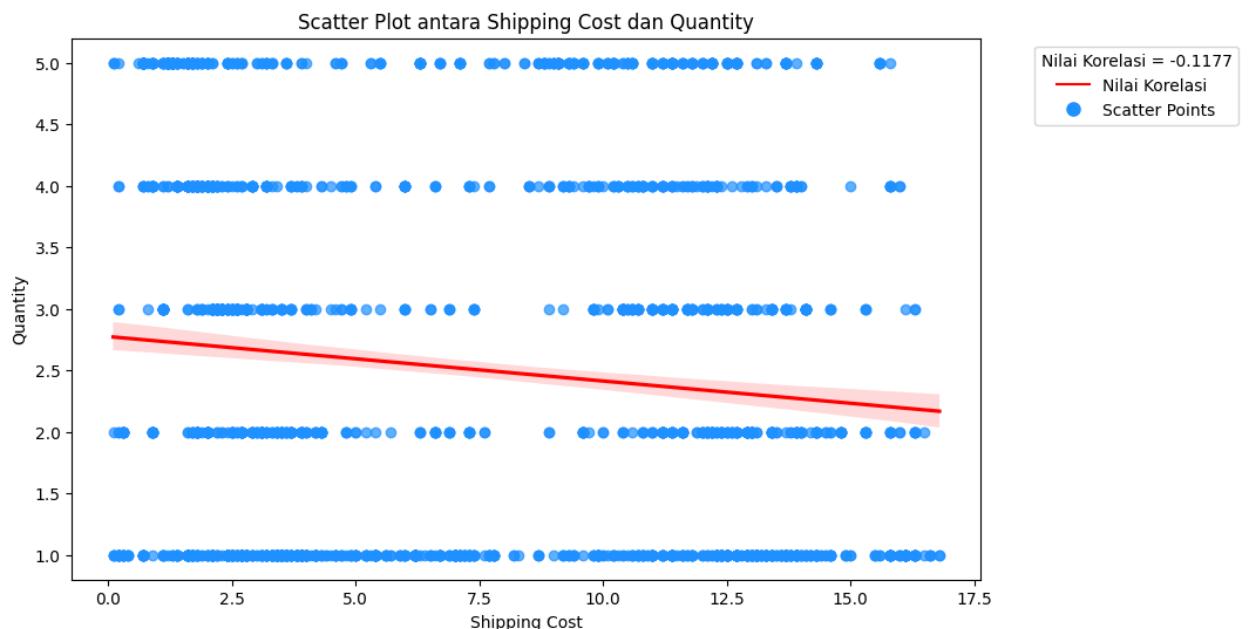
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Shipping Cost dan Quantity')
plt.xlabel('Shipping Cost')
plt.ylabel('Quantity')

plt.show()

```

Nilai korelasinya adalah -0.1177221024360056
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena koefisien korelasinya mendekati 0.

C. Profit

1. Korelasi antara Profit dengan Shipping Cost

```

from matplotlib.lines import Line2D

correlation = df['Profit'].corr(df['Shipping_Cost'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Profit', y='Shipping_Cost', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')
]

plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Profit dan Shipping Cost')
plt.xlabel('Profit')
plt.ylabel('Shipping Cost')

plt.show()

Nilai korelasinya adalah 0.9999814528852782
Kedua nilai berkorelasi lurus

```



Kedua atribut berkorelasi karena memiliki koefisien korelasi mendekati 1.

2. Korelasi antara Profit dengan Shipping Cost

```

from matplotlib.lines import Line2D

correlation = df['Profit'].corr(df['Aging'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Profit', y='Aging', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

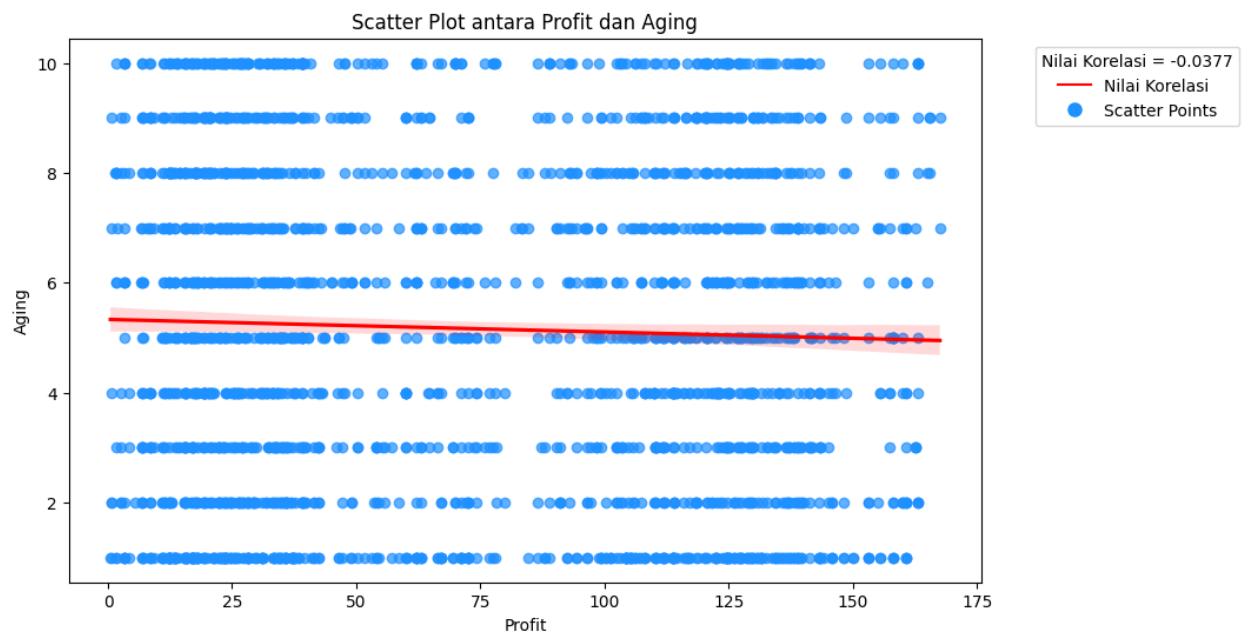
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
    
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Profit dan Aging')
plt.xlabel('Profit')
plt.ylabel('Aging')

plt.show()

```

Nilai korelasinya adalah -0.037719141294457664
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

3. Korelasi antara Profit dengan Sales

```

from matplotlib.lines import Line2D

correlation = df['Profit'].corr(df['Sales'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Profit', y='Sales', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

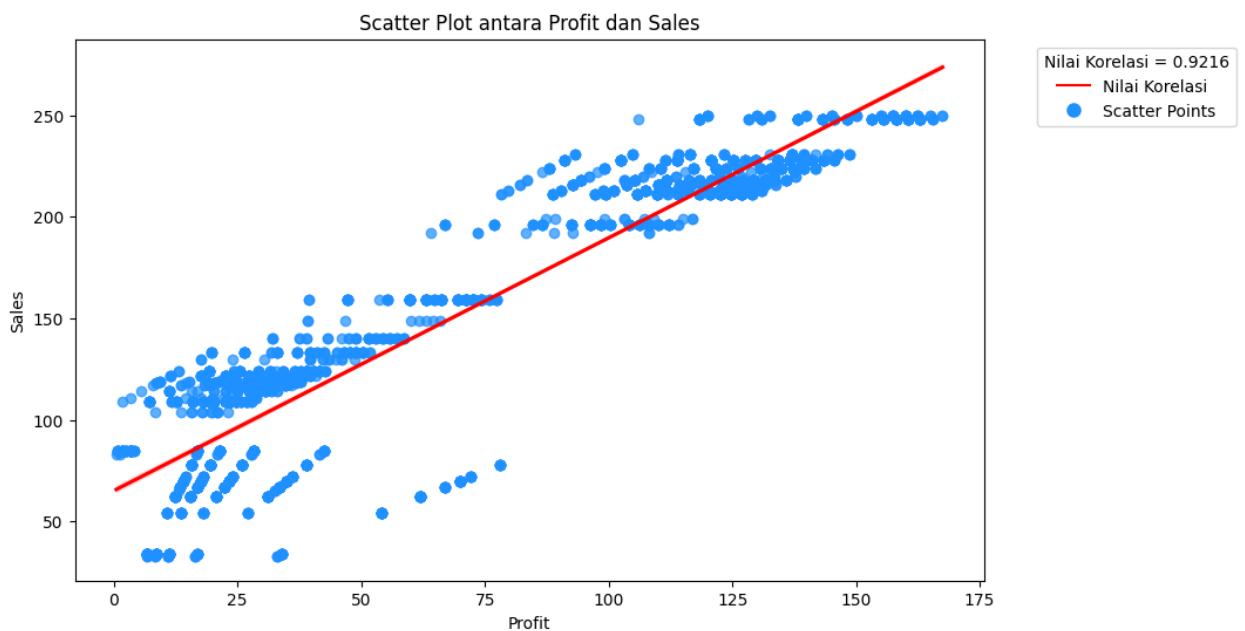
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
    
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Profit dan Sales')
plt.xlabel('Profit')
plt.ylabel('Sales')

plt.show()

```

Nilai korelasinya adalah 0.921616477282559
Kedua nilai berkorelasi lurus



Kedua atribut berkorelasi karena memiliki koefisien korelasi mendekati 1.

4. Korelasi antara Profit dengan Discount

```

from matplotlib.lines import Line2D

correlation = df['Profit'].corr(df['Discount'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Profit', y='Discount', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')
]

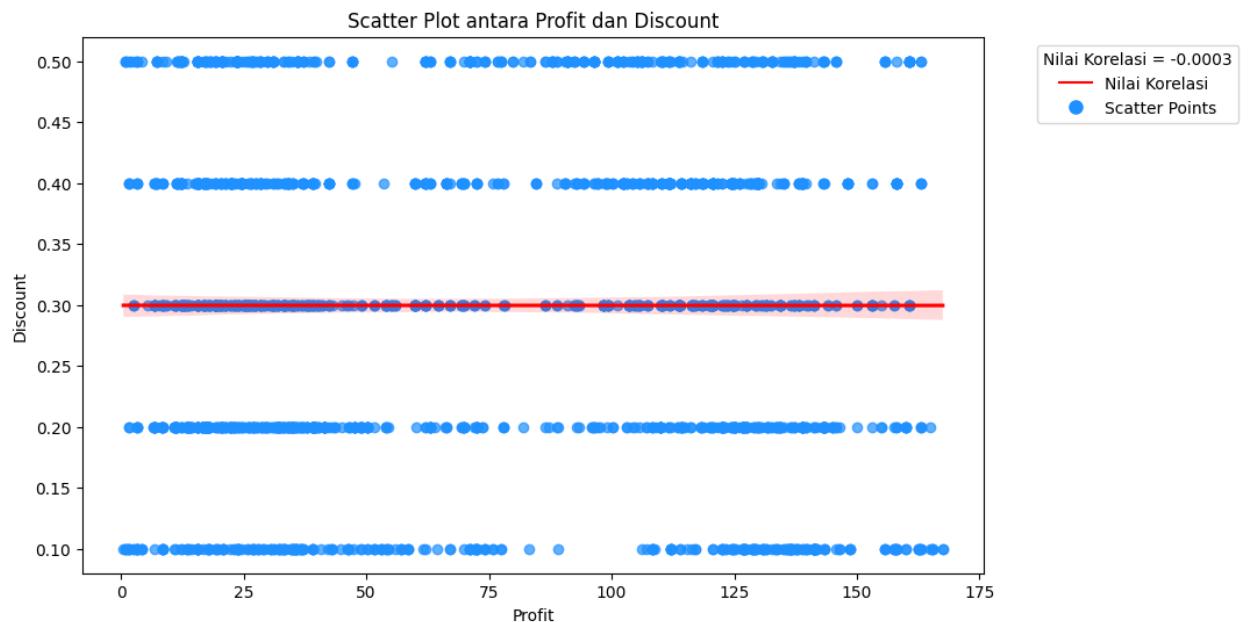
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Profit dan Discount')
plt.xlabel('Profit')
plt.ylabel('Discount')

plt.show()

Nilai korelasinya adalah -0.00025232827683302456
Kedua nilai tidak berkorelasi

```



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

5. Korelasi antara Profit dengan Quantity

```

from matplotlib.lines import Line2D

correlation = df['Profit'].corr(df['Quantity'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Profit', y='Quantity', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
]

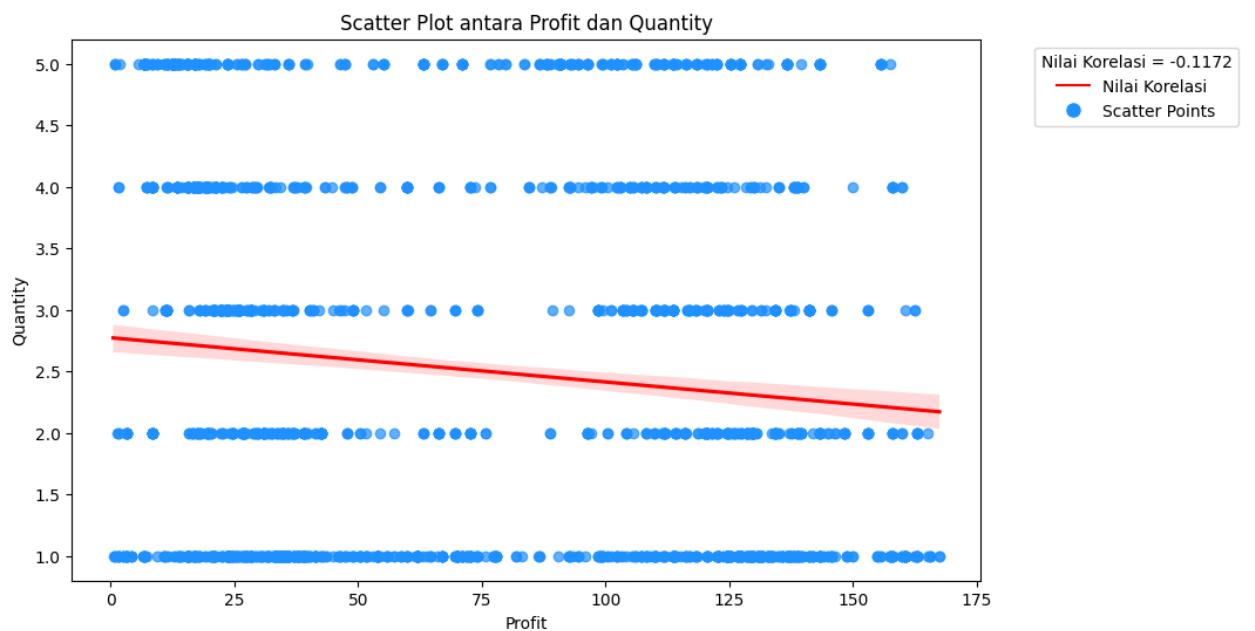
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Profit dan Quantity')
plt.xlabel('Profit')
plt.ylabel('Quantity')

plt.show()

```

Nilai korelasinya adalah -0.1171856406886578
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

D. Sales

1. Korelasi antara Sales dengan Aging

```

from matplotlib.lines import Line2D

correlation = df['Sales'].corr(df['Aging'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Sales', y='Aging', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

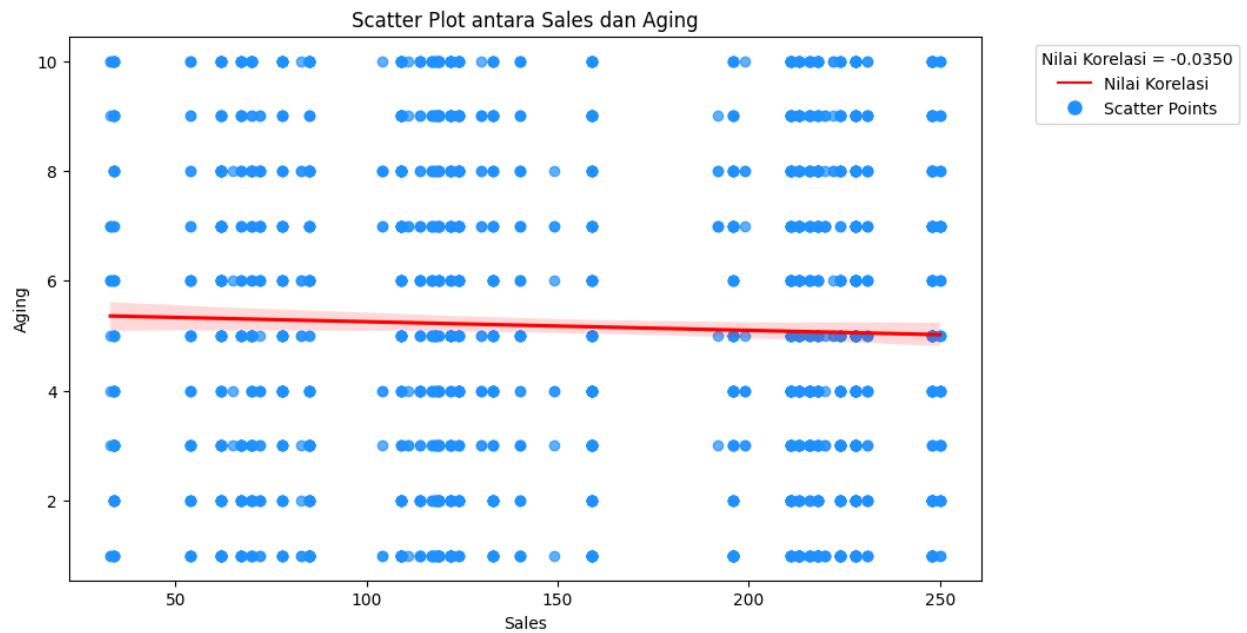
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Sales dan Aging')
plt.xlabel('Sales')
plt.ylabel('Aging')

plt.show()

```

Nilai korelasinya adalah -0.03496045938464408
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

2. Korelasi antara Sales dengan Shipping Cost

```

from matplotlib.lines import Line2D

correlation = df['Sales'].corr(df['Shipping_Cost'])
print(f'Nilai korelasinya adalah {correlation}')

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Sales', y='Shipping_Cost', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
]

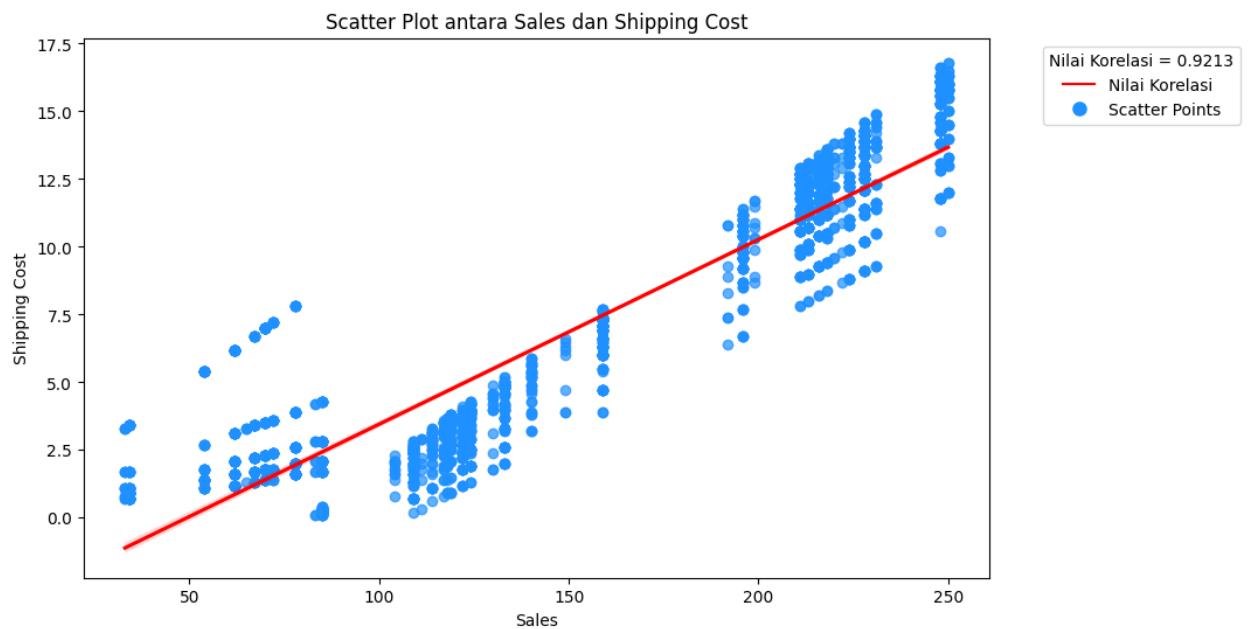
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Sales dan Shipping Cost')
plt.xlabel('Sales')
plt.ylabel('Shipping Cost')

plt.show()

```

Nilai korelasinya adalah 0.9213054279092486
Kedua nilai berkorelasi lurus



Kedua atribut berkorelasi karena memiliki koefisien korelasi mendekati 1.

3. Korelasi antara Sales dan Profit

```

from matplotlib.lines import Line2D

correlation = df['Sales'].corr(df['Profit'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Sales', y='Profit', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

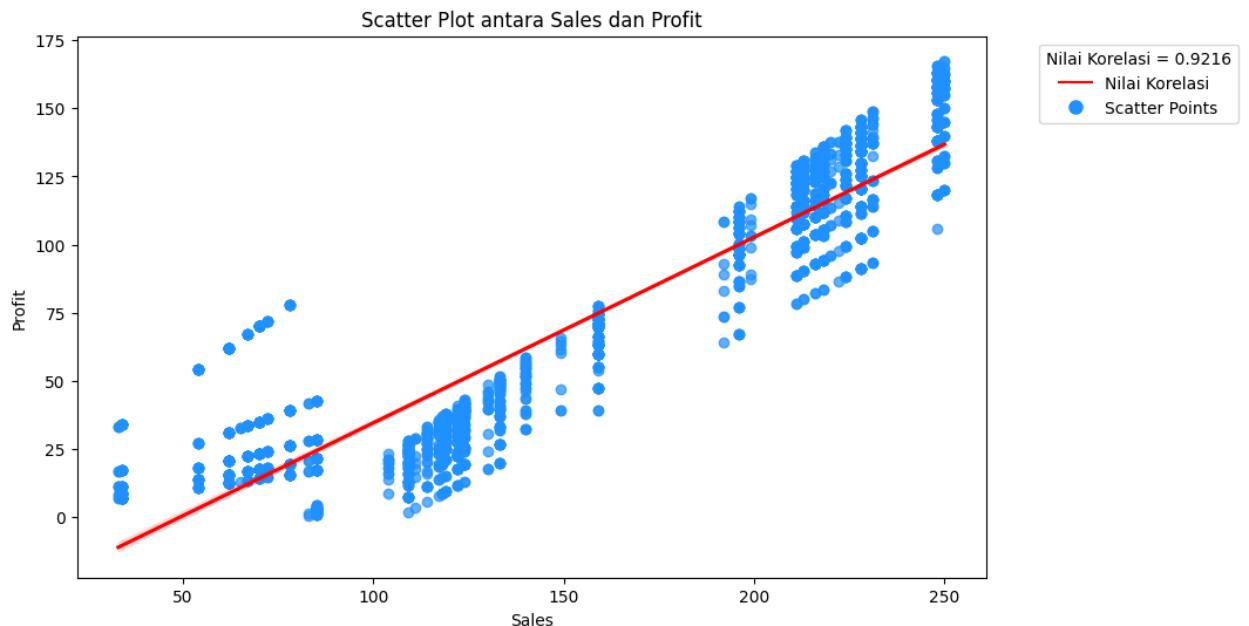
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Sales dan Profit')
plt.xlabel('Sales')
plt.ylabel('Profit')

plt.show()

Nilai korelasinya adalah 0.921616477282559
Kedua nilai berkorelasi lurus

```



Kedua atribut berkorelasi karena memiliki koefisien korelasi mendekati 1.

4. Korelasi antara Sales dengan Discount

```

from matplotlib.lines import Line2D

correlation = df['Sales'].corr(df['Discount'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Sales', y='Discount', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]

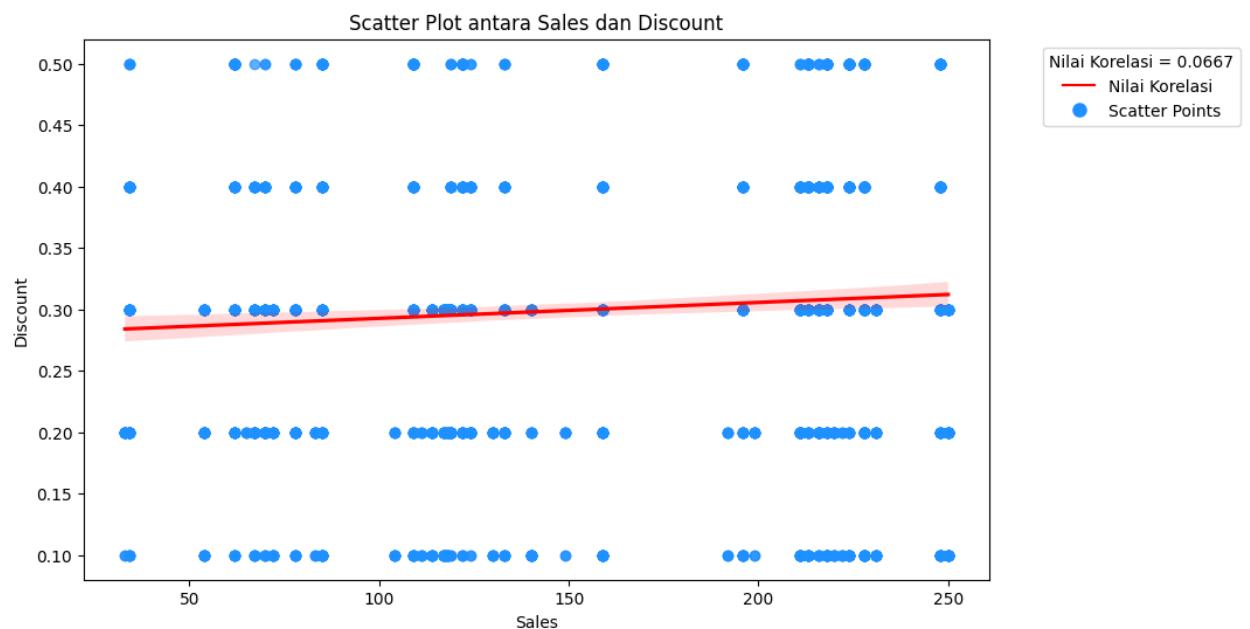
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Sales dan Discount')
plt.xlabel('Sales')
plt.ylabel('Discount')

plt.show()

```

Nilai korelasinya adalah 0.06674335557586303
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

5. Korelasi antara Sales dengan Quantity

```

from matplotlib.lines import Line2D

correlation = df['Sales'].corr(df['Quantity'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Sales', y='Quantity', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

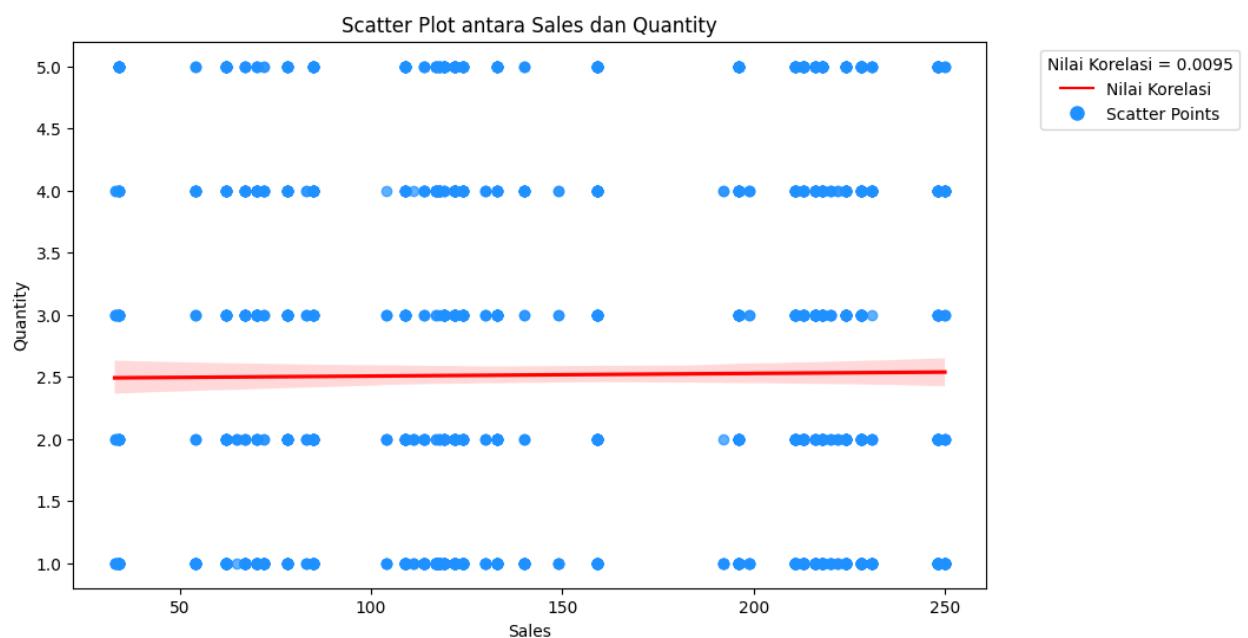
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
    
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Sales dan Quantity')
plt.xlabel('Sales')
plt.ylabel('Quantity')

plt.show()

Nilai korelasinya adalah 0.009476315521190633
Kedua nilai tidak berkorelasi

```



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

E. Discount

1. Korelasi antara Discount dengan Aging

```

from matplotlib.lines import Line2D

correlation = df['Discount'].corr(df['Aging'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Discount', y='Aging', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
]

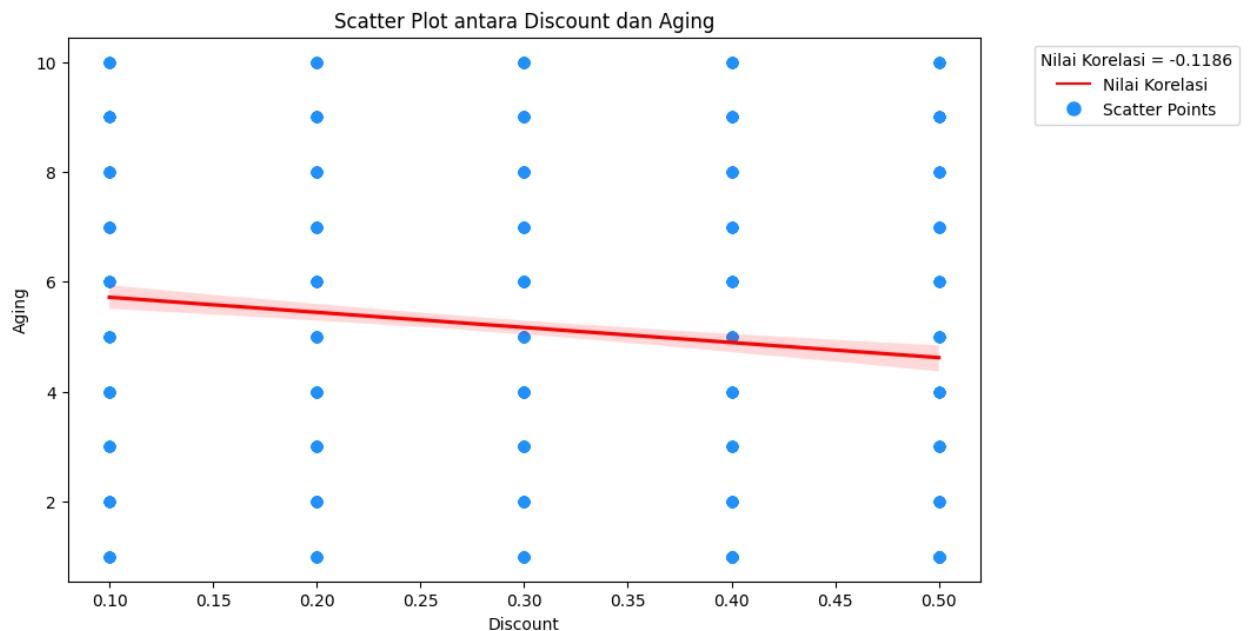
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Discount dan Aging')
plt.xlabel('Discount')
plt.ylabel('Aging')

plt.show()

Nilai korelasinya adalah -0.11857193323844717
Kedua nilai tidak berkorelasi

```



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0..

2. Korelasi antara Discount dengan Shipping Cost

```

from matplotlib.lines import Line2D

correlation = df['Discount'].corr(df['Shipping_Cost'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.replot(x='Discount', y='Shipping_Cost', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')
]

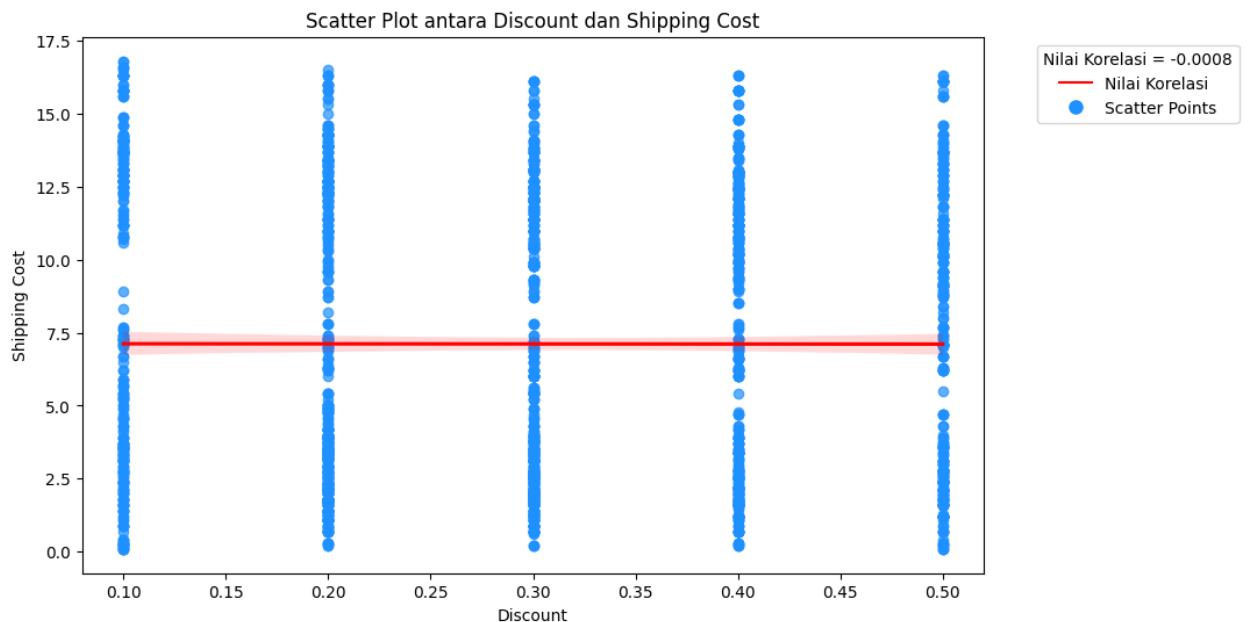
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Discount dan Shipping Cost')
plt.xlabel('Discount')
plt.ylabel('Shipping Cost')

plt.show()

```

Nilai korelasinya adalah -0.0007903480762710723
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

3. Korelasi antara Discount dengan Profit

```

from matplotlib.lines import Line2D

correlation = df['Discount'].corr(df['Profit'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Discount', y='Profit', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')
]

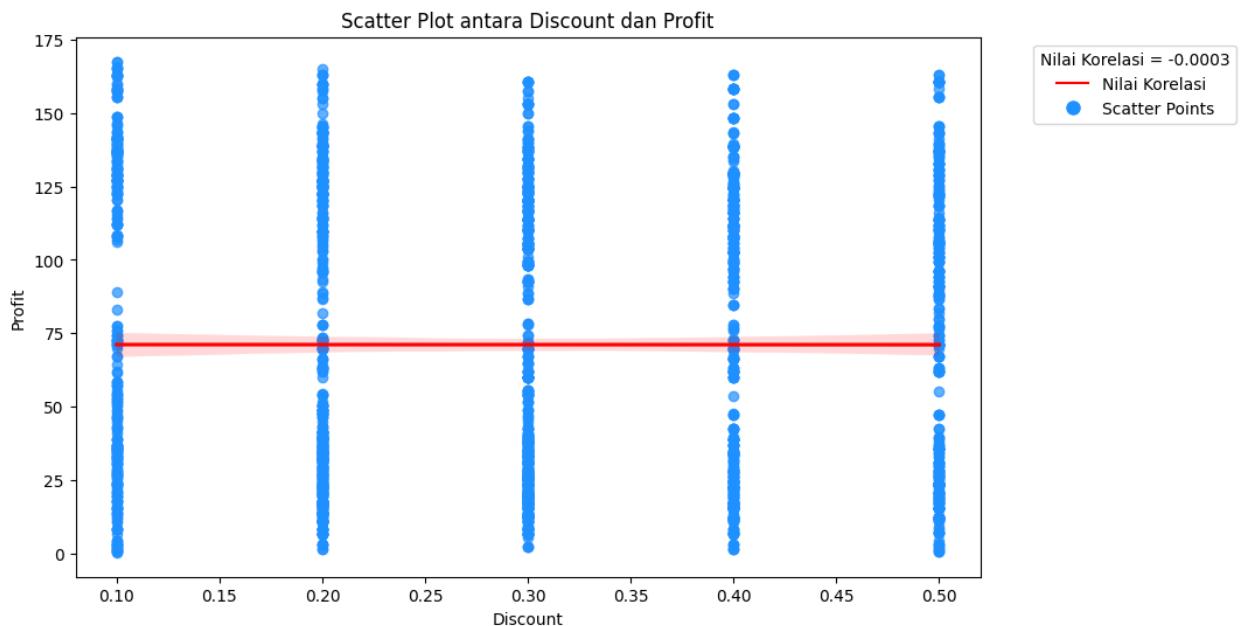
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Discount dan Profit')
plt.xlabel('Discount')
plt.ylabel('Profit')

plt.show()

Nilai korelasinya adalah -0.00025232827683302456
Kedua nilai tidak berkorelasi

```



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

4. Korelasi antara Discount dengan Sales

```

from matplotlib.lines import Line2D

correlation = df['Discount'].corr(df['Sales'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Discount', y='Sales', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')
]

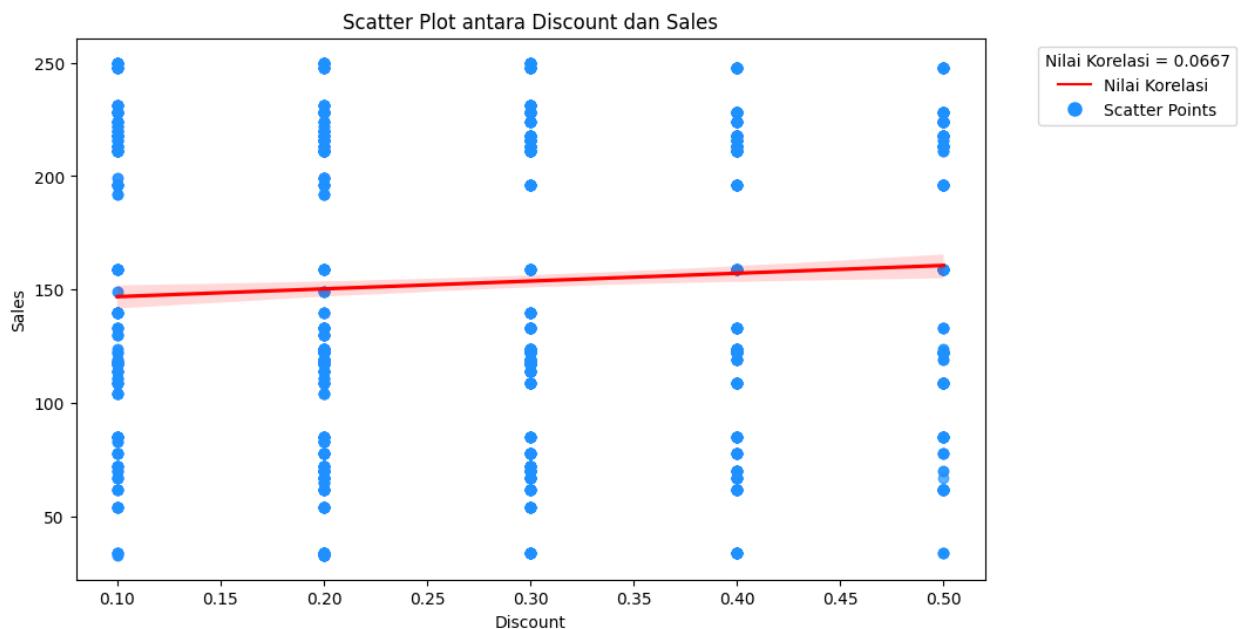
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Discount dan Sales')
plt.xlabel('Discount')
plt.ylabel('Sales')

plt.show()

```

Nilai korelasinya adalah 0.06674335557586303
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

5. Korelasi antara Discount dengan Quantity

```

from matplotlib.lines import Line2D

correlation = df['Discount'].corr(df['Quantity'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Discount', y='Quantity', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')
]

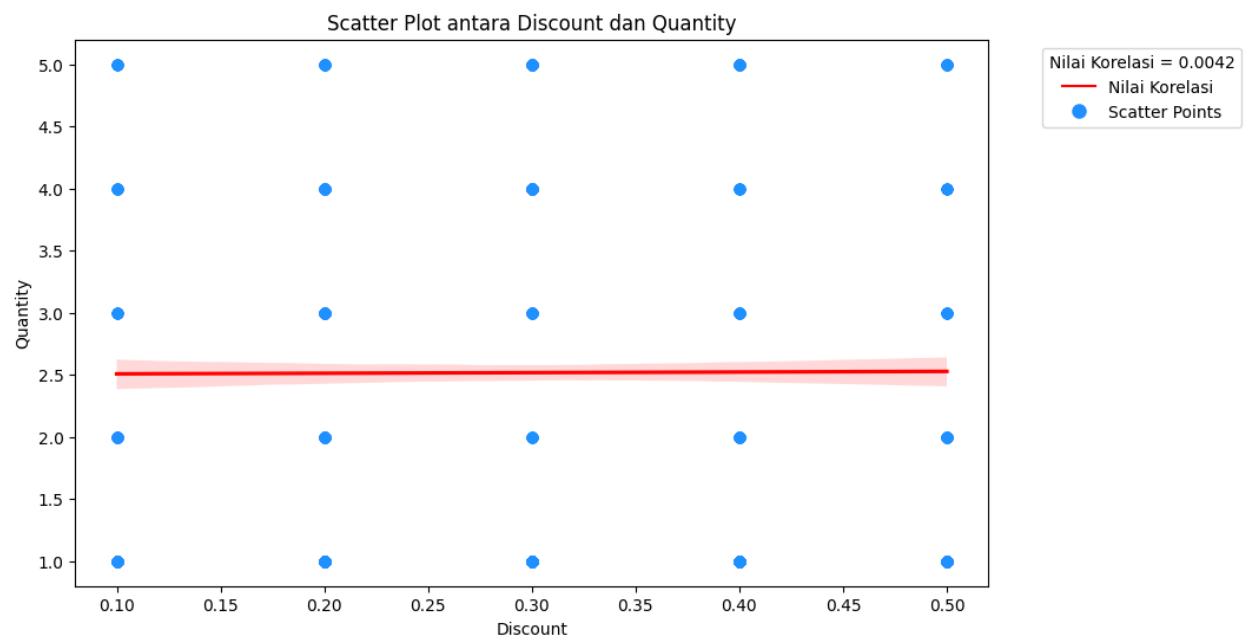
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Discount dan Quantity')
plt.xlabel('Discount')
plt.ylabel('Quantity')

plt.show()

```

Nilai korelasinya adalah 0.004209170932839911
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

F. Quantity

1. Korelasi antara Quantity dengan Aging

```

from matplotlib.lines import Line2D

correlation = df['Quantity'].corr(df['Aging'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.replot(x='Quantity', y='Aging', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

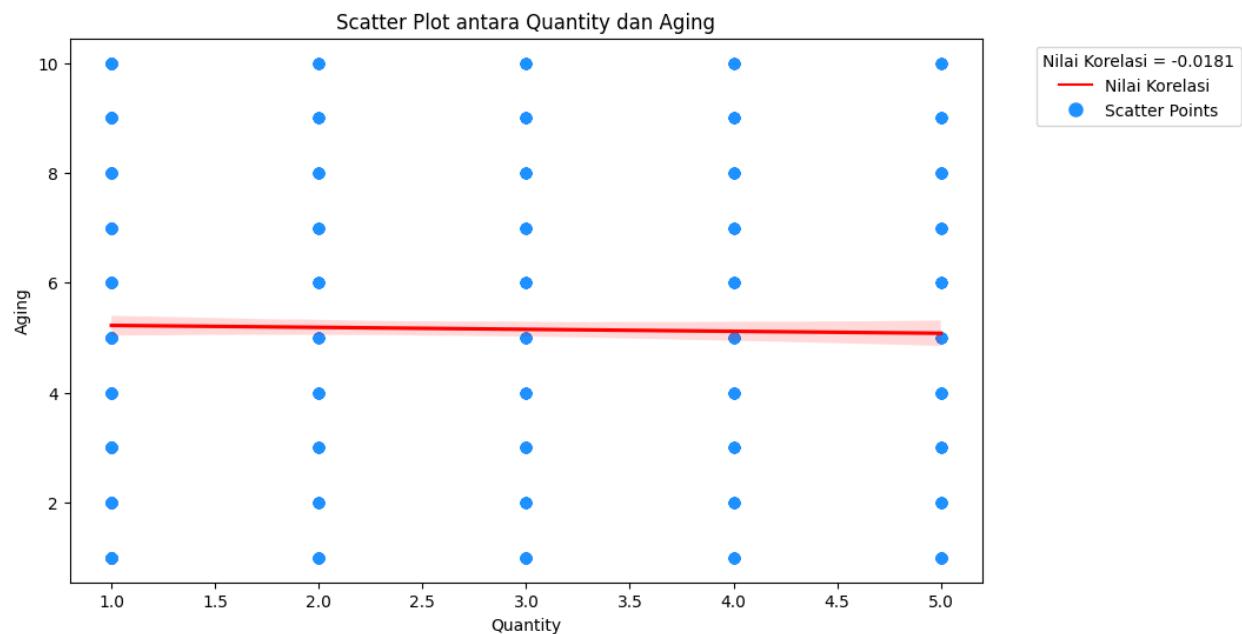
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]

plt.legend(handles=legend_elements, title='Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))
plt.title('Scatter Plot antara Quantity dan Aging')
plt.xlabel('Quantity')
plt.ylabel('Aging')

plt.show()

```

Nilai korelasinya adalah -0.01809719299097352
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

2. Korelasi antara Quantity dengan Shipping Cost

```

from matplotlib.lines import Line2D

correlation = df['Quantity'].corr(df['Shipping_Cost'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Quantity', y='Shipping_Cost', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]

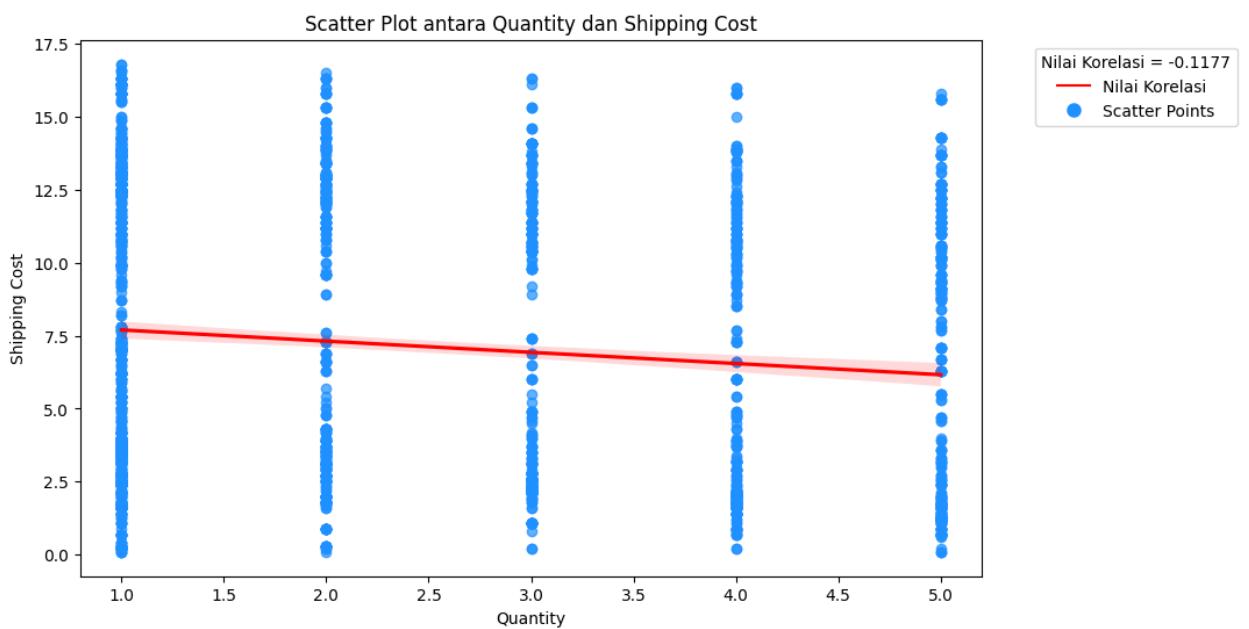
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Quantity dan Shipping Cost')
plt.xlabel('Quantity')
plt.ylabel('Shipping Cost')

plt.show()

```

Nilai korelasinya adalah -0.11772210243600562
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

3. Korelasi antara Quantity dengan Profit

```

from matplotlib.lines import Line2D

correlation = df['Quantity'].corr(df['Profit'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Quantity', y='Profit', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

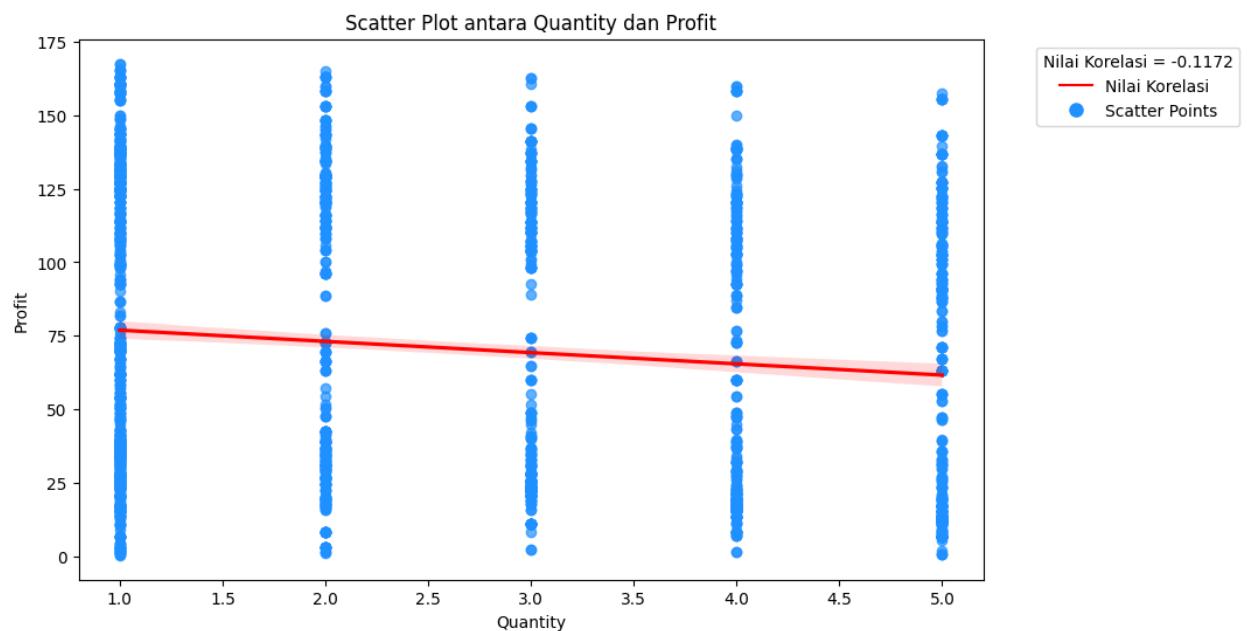
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
    
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Quantity dan Profit')
plt.xlabel('Quantity')
plt.ylabel('Profit')

plt.show()

```

Nilai korelasinya adalah -0.1171856406886578
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

4. Korelasi antara Quantity dengan Sales

```

from matplotlib.lines import Line2D

correlation = df['Quantity'].corr(df['Sales'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Quantity', y='Sales', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]

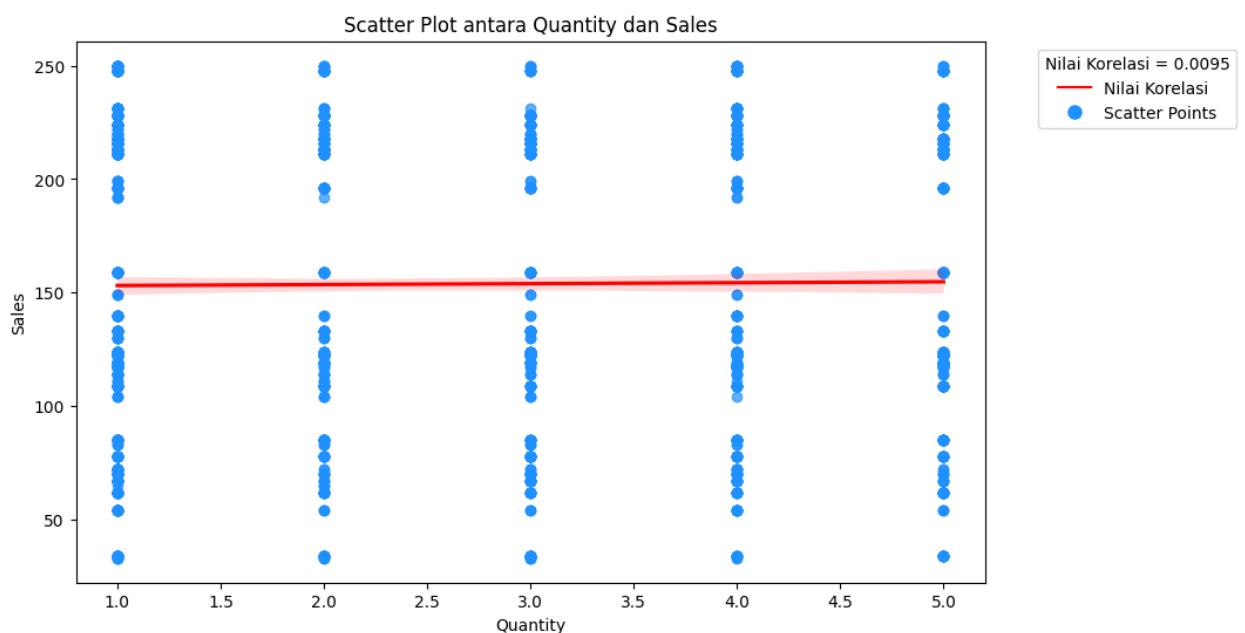
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Quantity dan Sales')
plt.xlabel('Quantity')
plt.ylabel('Sales')

plt.show()

```

Nilai korelasinya adalah 0.009476315521190631
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

5. Korelasi antara Quantity dengan Discount

```

from matplotlib.lines import Line2D

correlation = df['Quantity'].corr(df['Discount'])
print(f"Nilai korelasinya adalah {correlation}")

if 0.5 < correlation <= 1:
    print("Kedua nilai berkorelasi lurus")
elif -0.5 < correlation < 0.5:
    print("Kedua nilai tidak berkorelasi")
elif -1 <= correlation < -0.5:
    print("Kedua kolom berkorelasi terbalik")

plt.figure(figsize=(10, 6))
sns.regplot(x='Quantity', y='Discount', data=df, color='red', scatter_kws={'alpha': 0.7, 'color': 'dodgerblue'}, label='Regression Line')

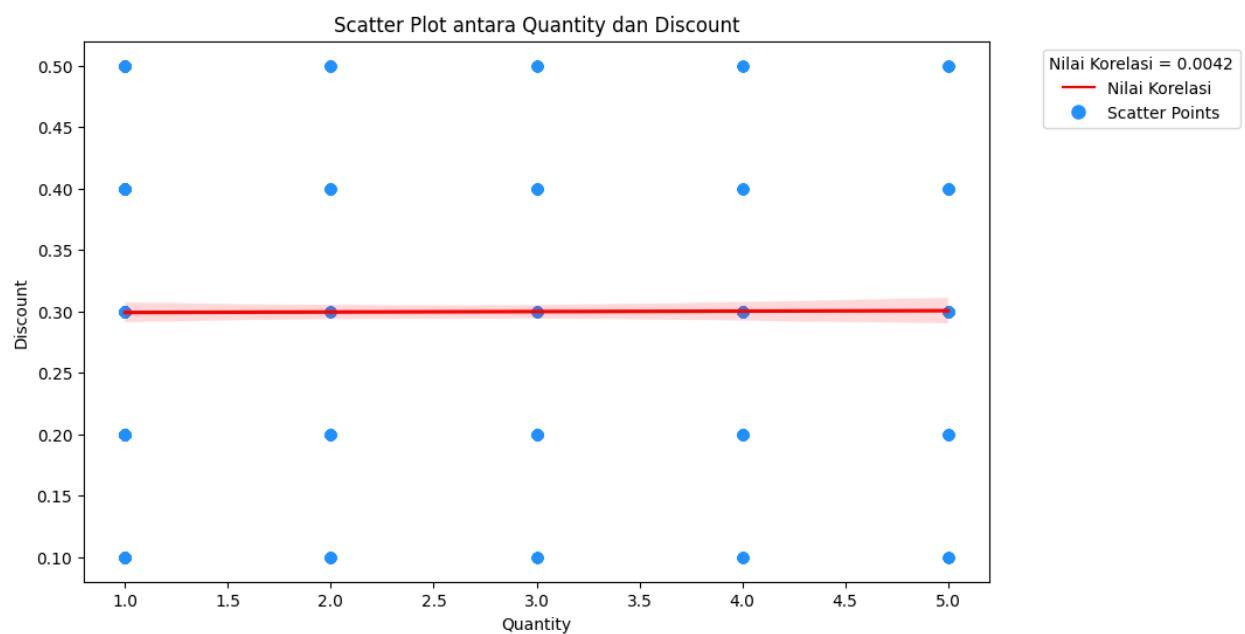
legend_elements = [
    Line2D([0], [0], color='red', label='Nilai Korelasi'),
    Line2D([0], [0], marker='o', color='dodgerblue', label='Scatter Points', markersize=8, linestyle='None')]
plt.legend(handles=legend_elements, title=f'Nilai Korelasi = {correlation:.4f}', bbox_to_anchor=(1.05, 1))

plt.title('Scatter Plot antara Quantity dan Discount')
plt.xlabel('Quantity')
plt.ylabel('Discount')

plt.show()

```

Nilai korelasinya adalah 0.004209170932839912
Kedua nilai tidak berkorelasi



Kedua atribut tidak berkorelasi karena memiliki koefisien korelasi mendekati 0.

BAB IV

PENUTUP

3.1. Kesimpulan

Berdasarkan analisis data yang dipilih, dapat disimpulkan data yang masih belum diolah sangat sulit untuk diolah dan digunakan kembali. Visualisasi data sangat diperlukan untuk mencapai tingkat efisiensi yang tinggi dalam mengolah data. Dalam data set “E-com dataset dari Januari 2018 sampai dengan Desember 2018” yaitu total (sum), rata-rata (mean), penjualan terbesar dan terkecil (max dan min), dan persentil. Analisis yang dilakukan akan membahas dalam beberapa variabel, yakni *product*, dan *customer* dengan fokus utama adalah *product*

Berdasarkan hasil dari analisis data yang dilakukan didapatkan total penjualan untuk setiap *product*, yaitu; *Fashion* dengan total 1053 pieces, *Home and Furniture* dengan 664 pieces, dan *Home and Accessories* dengan total 327 pieces, dan *Electronic* dengan total 117 pieces. Penjualan terbanyak dihasilkan dari produk *Fossil Watch* dengan kuantitas 111 pieces.

Korelasi dari penjualan setiap kategori memiliki variasi yakni; pertama *aging* untuk produk secara keseluruhan dengan *shipping cost* tidak berkorelasi. Kedua *aging* dengan *profit* untuk produk secara keseluruhan tidak berkorelasi. Ketiga *aging* dengan *sales* keduanya tidak berkorelasi. Keempat *aging* dengan *discount* keduanya tidak berkorelasi. Kelima *aging* dengan *quantity* untuk setiap barang tidak berkorelasi. Keenam *shipping cost* dengan *profit* kedua nilai berkorelasi lurus. Ketujuh *shipping cost* dengan *discount* kedua nilai tidak berkorelasi. Kedelapan *shipping cost* dengan *sales* keduanya tidak berkorelasi. Kesembilan *shipping cost* dengan *quantity* kedua nilai tidak berkorelasi. Kesepuluh *profit* dengan *sales* kedua nilai berkorelasi lurus. Kesebelas *profit* dengan *discount* kedua nilai tidak berkorelasi. Keduabelas *profit* dengan *quantity* kedua nilai tidak berkorelasi. Ketigabelas *sales* dengan *discount* kedua nilai tidak berkorelasi. Keempatbelas *sales* dengan *quantity* kedua nilai tidak berkorelasi.

3.2. Pembagian Tugas

Nama	NIM	Tugas
Audy Alicia Renatha Tirayoh	19623169	Laporan, Powerpoint
Nicholas Andhika Lucas	19623218	Deskripsi dan Karakteristik Data, Statistik, Visualisasi, Korelasi
Amanda Nadine Calysta	16523204	Deskripsi dan Karakteristik Data, Laporan
Lintang Suminar	16523162	Statistik, Visualisasi, Korelasi, Laporan, Powerpoint
Reinhard Iven Winnata	16523071	Data cleansing, Laporan

SUMBER

- Menekse, Merve. 2021. *E-Commerce Dataset*. Online:
[\(https://www.kaggle.com/datasets/mervemenekse/ecommerce-dataset/data\)](https://www.kaggle.com/datasets/mervemenekse/ecommerce-dataset/data)
Tim Materi Pengenalan Komputasi. 2023. *Modul Praktikum Pengenalan Komputasi*. Online