

# aws-analytics

October 7, 2023

## 1 Amazon Web Service

Analyzed By : Ali Wafa Afif Ramdhan

---

### Latar Belakang

**Amazon Web Services (AWS)** adalah anak perusahaan Amazon.com yang telah mendominasi industri **layanan komputasi awan** sejak berdiri pada tahun 2006.

Dataset ini memberikan kita wawasan tentang penjualan AWS kepada perusahaan lain dalam lingkup **B2B (Business-to-Business)** dari tahun 2020 hingga 2023.

Dataset ini bisa di Akses di : [Link Data Set](#)

**Layanan AWS** mencakup beragam aspek, termasuk:

1. **Storage:** Penyimpanan data dalam berbagai format dan kapasitas.
2. **Database:** Layanan basis data yang aman dan skalabel.
3. **Security:** Keamanan data dan infrastruktur yang sangat penting.
4. **Messaging:** Komunikasi dan pertukaran informasi yang efisien.
5. **Compute:** Sumber daya komputasi yang fleksibel dan kuat.

**Segment produk AWS** yang ditawarkan kepada pelanggan B2B juga bervariasi:

1. **SMB (Small and Medium-sized Business):** AWS menyediakan solusi dengan biaya terjangkau untuk bisnis kecil dan menengah, termasuk mesin virtual dan basis data.
2. **Strategic:** Ini adalah solusi untuk bisnis yang telah berkembang dan memerlukan layanan lebih canggih, seperti manajemen aplikasi web dan analitik data.
3. **Enterprise:** AWS menawarkan solusi lengkap untuk bisnis skala besar yang memerlukan tingkat keamanan, dukungan, dan fitur tingkat tinggi, seperti manajemen hak akses dan dukungan khusus.

Dengan demikian, AWS bukan hanya pemimpin dalam industri komputasi awan, tetapi juga mitra yang dapat memenuhi beragam kebutuhan bisnis B2B dengan berbagai solusi yang tersedia.

### 1.1 Pernyataan Masalah Dan Output Yang di Harapkan

Amazon Web Services (AWS) memiliki komitmen untuk mencapai pertumbuhan setiap tahunnya, tetapi dihadapkan pada sejumlah kendala yang perlu diatasi. Kendala-kendala ini meliputi:

1. **Profitabilitas Produk:** AWS menghadapi tantangan dalam mengoptimalkan profitabilitas beberapa produknya yang saat ini mengalami kerugian.
2. **Penurunan Penjualan dan Profit:** Terjadi penurunan dalam penjualan dan profit yang memerlukan perhatian khusus untuk memahami penyebabnya dan mengambil tindakan yang sesuai.
3. **Perencanaan Pemasaran yang Tidak Terukur:** AWS perlu meningkatkan perencanaan pemasaran yang lebih terukur, termasuk waktu pelaksanaan, lokasi, dan target pasar.
4. **Rencana Diskon yang Tidak Terukur:** Rencana diskon yang ada belum cukup terukur dalam hal waktu, lokasi, dan target pasar, sehingga perlu diperbaiki.

Dalam rangka mencapai tujuan ini, analisis akan difokuskan pada beberapa variabel kunci:

1. **Evaluasi Penjualan dan Profit Menurut Industri, Segmen, Asal (Negara dan Kota), serta Produk:** Dengan memahami kinerja penjualan dan profitabilitas dalam berbagai industri, segmen, asal (negara dan kota), serta produk, kita dapat mengidentifikasi area yang memerlukan perbaikan dan mengalokasikan sumber daya secara efisien.
2. **Analisis Tren Penjualan dan Profitabilitas Tahunan dan Bulanan:** Melalui analisis tren penjualan dan profitabilitas secara tahunan dan bulanan, kita dapat mengidentifikasi pola dan sinyal peringatan yang memungkinkan pengambilan tindakan yang tepat pada waktu yang tepat.
3. **Dampak Diskon terhadap Penjualan dan Profit:** Dengan mengevaluasi dampak diskon terhadap penjualan dan profitabilitas, kita dapat mengembangkan rencana diskon yang lebih efektif dengan memperhatikan waktu, lokasi, dan target pasar.

Melalui analisis mendalam ini dan pengembangan strategi yang tepat, AWS dapat mengatasi kendala yang dihadapi dan merencanakan langkah-langkah yang akan membawa pertumbuhan berkelanjutan dalam penjualan, profitabilitas, dan pemasaran produk B2B ke depan.

---

Stakeholder: **Tim Pemasaran dan Tim Pengembangan Produk (R&D)**

## 2 Data Info,Cleaning dan Testing

Tahap awal kita akan melakukan tahap Data Information dan Data Cleaning yang sangat penting. Ini membantu kita memahami data yang ada, jenis data yang digunakan, serta mencari potensi masalah atau anomali dalam dataset kita. Selain itu, kita juga akan menggunakan statistik untuk mendapatkan informasi yang lebih dalam.

Tahap ini membantu kita:

1. **Memahami Data:** Untuk tahu apa yang ada dalam data kita, seperti jenis informasi apa yang kita miliki.
2. **Identifikasi Tipe Data:** Untuk mengetahui apakah data itu angka, kategori, atau tanggal, yang membantu kita merencanakan analisis lebih lanjut.
3. **Deteksi Masalah:** Untuk menemukan masalah dalam data yang perlu diperbaiki agar analisis kita akurat.

4. **Analisis Statistik:** Melakukan perhitungan statistik sederhana untuk melihat tren atau pola dalam data.

Dengan melakukan langkah-langkah ini, kita akan siap untuk menganalisis data dengan lebih baik dan membuat rencana yang lebih tepat untuk meningkatkan kinerja AWS dalam hal penjualan, profitabilitas, dan penjualan produk B2B.

## 2.1 Import Data and Data Info

### 2.1.1 Import Data

Pada tahap ini kita akan import library yang kita perlukan

```
[1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import scipy.stats as ss
from scipy.stats import kruskal
from scipy import stats

warnings.filterwarnings('ignore')
```

Pada tahap ini kita memasukan Data Set Amazon Web Service

```
[2]: AWS= pd.read_csv('Amazon Web Service Sales.csv')
AWS
```

```
[2]:
```

	Row ID	Order ID	Order Date	Date Key	Contact Name \
0	1	EMEA-2022-152156	11/9/2022	20221109	Nathan Bell
1	2	EMEA-2022-152156	11/9/2022	20221109	Nathan Bell
2	3	AMER-2022-138688	6/13/2022	20220613	Deirdre Bailey
3	4	EMEA-2021-108966	10/11/2021	20211011	Zoe Hodges
4	5	EMEA-2021-108966	10/11/2021	20211011	Zoe Hodges
...	...	...	...	...	...
9989	9990	EMEA-2020-110422	1/22/2020	20200122	Dan Allan
9990	9991	AMER-2023-121258	2/27/2023	20230227	Anthony Sanderson
9991	9992	AMER-2023-121258	2/27/2023	20230227	Anthony Sanderson
9992	9993	AMER-2023-121258	2/27/2023	20230227	Anthony Sanderson
9993	9994	AMER-2023-119914	5/5/2023	20230505	Jasmine Springer

	Country	City	Region	Subregion	Customer \
0	Ireland	Dublin	EMEA	UKIR	Chevron
1	Ireland	Dublin	EMEA	UKIR	Chevron
2	United States	New York City	AMER	NAMER	Phillips 66
3	Germany	Stuttgart	EMEA	EU-WEST	Royal Dutch Shell
4	Germany	Stuttgart	EMEA	EU-WEST	Royal Dutch Shell
...	...	...	...	...	...

9989	Germany	Hamburg	EMEA	EU-WEST	Johnson & Johnson
9990	United States	Milwaukee	AMER	NAMER	American Express
9991	United States	Milwaukee	AMER	NAMER	American Express
9992	United States	Milwaukee	AMER	NAMER	American Express
9993	United States	Dallas	AMER	NAMER	Comcast

	Customer ID	Industry	Segment	Product \
0	1017	Energy	SMB	Marketing Suite
1	1017	Energy	SMB	FinanceHub
2	1056	Energy	Strategic	FinanceHub
3	1031	Energy	SMB	ContactMatcher
4	1031	Energy	SMB	Marketing Suite - Gold
...	...	...	...	...
9989	1055	Healthcare	SMB	SaaS Connector Pack
9990	1005	Finance	SMB	SaaS Connector Pack
9991	1005	Finance	SMB	Site Analytics
9992	1005	Finance	SMB	Support
9993	1022	Communications	SMB	OneView

	License	Sales	Quantity	Discount	Profit
0	16GRM07R1K	261.9600	2	0.00	41.9136
1	QLIW57KZUV	731.9400	3	0.00	219.5820
2	JI6BVL70HQ	14.6200	2	0.00	6.8714
3	DE9GJKGD44	957.5775	5	0.45	-383.0310
4	OIF7NY23WD	22.3680	2	0.20	2.5164
...	...	...	...	...	...
9989	ETHXMB5TMD	25.2480	3	0.20	4.1028
9990	LYHOKTRY7L	91.9600	2	0.00	15.6332
9991	I5SVM16M88	258.5760	2	0.20	19.3932
9992	JHLW4P6LQ2	29.6000	4	0.00	13.3200
9993	3NTBV9B3PS	243.1600	2	0.00	72.9480

[9994 rows x 19 columns]

Berikut adalah penjelasan lengkap dari Dataset kita:

Nama Kolom	Keterangan
Row ID	ID unik setiap pesanan
Order ID	ID unik setiap pesanan
Order Date	Tanggal ketika pesanan dilakukan
Date Key	Representasi numerik dari tanggal pesanan (YYYYMMDD)
Contact Name	Nama orang yang melakukan pemesanan
Country	Negara tempat pemesanan dilakukan
City	Kota tempat pemesanan dilakukan
Region	Wilayah tempat pesanan dilakukan
Subregion	Subwilayah tempat pesanan dilakukan
Customer	Nama perusahaan yang melakukan pemesanan

Nama Kolom	Keterangan
Customer ID	ID unik untuk setiap pelanggan
Industry	Industri tempat pelanggan berada
Segment	Segmen pelanggan
Product	Produk yang dipesan
Licence	ID lisensi unik untuk produk
Sales	Jumlah total penjualan untuk transaksi
Quantity	Jumlah total item dalam transaksi
Discount	Diskon yang diterapkan pada transaksi
Profit	Keuntungan dari transaksi

### 2.1.2 Data Info

Berikut adalah info data type dari setiap kolom nya

```
[3]: AWS.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 19 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Row ID          9994 non-null  int64
1   Order ID        9994 non-null  object
2   Order Date      9994 non-null  object
3   Date Key        9994 non-null  int64
4   Contact Name    9994 non-null  object
5   Country         9994 non-null  object
6   City            9994 non-null  object
7   Region          9994 non-null  object
8   Subregion       9994 non-null  object
9   Customer        9994 non-null  object
10  Customer ID     9994 non-null  int64
11  Industry        9994 non-null  object
12  Segment         9994 non-null  object
13  Product         9994 non-null  object
14  License         9994 non-null  object
15  Sales           9994 non-null  float64
16  Quantity        9994 non-null  int64
17  Discount        9994 non-null  float64
18  Profit          9994 non-null  float64
dtypes: float64(3), int64(4), object(12)
memory usage: 1.4+ MB
```

Semua data di sini sesuai dengan format yang diinginkan, kecuali kolom Order date masih dalam bentuk objek, yang nantinya akan diubah menjadi tipe data datetime.

Pada tahap ini, kita akan menggali informasi statistik dari data. Di sini, kita dapat melihat statistik untuk variabel seperti Sales, Quantity (Jumlah Item yang Terjual), Discount, dan Profit, termasuk nilai rata-rata (Mean), jumlah data (Count), nilai terkecil (Min), dan nilai terbesar (Max), serta statistik lainnya.

```
[4]: AWS.describe()
```

```
[4]:
```

	Row ID	Date Key	Customer ID	Sales	Quantity \
count	9994.000000	9.994000e+03	9994.000000	9994.000000	9994.000000
mean	4997.500000	2.021803e+07	1049.771963	229.858001	3.789574
std	2885.163629	1.123898e+04	29.719388	623.245101	2.225110
min	1.000000	2.020010e+07	1001.000000	0.444000	1.000000
25%	2499.250000	2.021052e+07	1024.000000	17.280000	2.000000
50%	4997.500000	2.022063e+07	1049.000000	54.490000	3.000000
75%	7495.750000	2.023052e+07	1076.000000	209.940000	5.000000
max	9994.000000	2.023123e+07	1101.000000	22638.480000	14.000000

	Discount	Profit
count	9994.000000	9994.000000
mean	0.156203	28.656896
std	0.206452	234.260108
min	0.000000	-6599.978000
25%	0.000000	1.728750
50%	0.200000	8.666500
75%	0.200000	29.364000
max	0.800000	8399.976000

Di sini, kita menghitung nilai minimum, maksimum, dan rata-rata dari kolom sales, quantity, discount, dan profit. Hal ini bermanfaat untuk mengidentifikasi potensi anomali dalam data tersebut.

## 2.2 Data Cleaning, Filtering and Converting

### 2.2.1 Data Duplicated

Pada tahap ini, kita akan memeriksa apakah terdapat data duplikat

```
[5]: #Duplicated Value - No Duplicate Value
AWS[AWS.duplicated()].sum()
```

```
[5]: Row ID      0.0
      Order ID   0.0
      Order Date 0.0
      Date Key   0.0
      Contact Name 0.0
      Country    0.0
      City       0.0
      Region     0.0
      Subregion  0.0
      Customer   0.0
```

```

Customer ID    0.0
Industry       0.0
Segment        0.0
Product        0.0
License        0.0
Sales          0.0
Quantity       0.0
Discount       0.0
Profit         0.0
dtype: float64

```

hasil pemeriksaan menunjukkan bahwa data ini tidak mengandung duplikasi.

### 2.2.2 Missing Value

Dalam tahap ini, kita akan melakukan pemeriksaan untuk mengidentifikasi apakah terdapat nilai yang hilang (missing value) dalam dataset.

```

[6]: #Missing Value - NO MISSING VALUE
AWS[AWS.isna()].sum()

```

```

[6]: Row ID          0.0
      Order ID       0
      Order Date     0
      Date Key       0.0
      Contact Name   0
      Country        0
      City           0
      Region         0
      Subregion      0
      Customer       0
      Customer ID    0.0
      Industry       0
      Segment        0
      Product        0
      License        0
      Sales          0.0
      Quantity       0.0
      Discount       0.0
      Profit         0.0
      dtype: object

```

Hasil pemeriksaan awal menunjukkan bahwa tidak ada nilai yang hilang dalam dataset ini.

Untuk memastikan bahwa tidak ada data yang hilang, kita akan melakukan uji khusus pada data Profit. Kita akan mencari apakah terdapat profit yang bernilai 0 dengan kondisi discount 0, karena tidak mungkin profit nya 0 jika tidak ada discount yang di terapkan.

```

[7]: AWS[(AWS['Profit']== 0) & (AWS['Discount']== 0)].count()

```

```
[7]: Row ID          30
      Order ID       30
      Order Date     30
      Date Key       30
      Contact Name    30
      Country         30
      City            30
      Region         30
      Subregion       30
      Customer        30
      Customer ID     30
      Industry        30
      Segment         30
      Product         30
      License         30
      Sales           30
      Quantity        30
      Discount        30
      Profit          30
      dtype: int64
```

Setelah mengetahui kondisi ini, kita akan mengonversi 30 baris data tersebut ke dalam bentuk persentase.

```
[8]: count_matching_rows = len(AWS[(AWS['Profit'] == 0) & (AWS['Discount'] == 0)])
      percentage = (count_matching_rows / len(AWS)) * 100
      percentage
```

```
[8]: 0.3001801080648389
```

Hasilnya menunjukkan bahwa nilainya hanya sebesar 0.3%.

Dikarenakan hanya 0.3%, kita akan menghapus data-data tersebut.

```
[9]: AWS.drop(AWS[(AWS['Profit'] == 0) & (AWS['Discount'] == 0)].index,inplace=True)
      AWS[(AWS['Profit']== 0) & (AWS['Discount']== 0)].count()
```

```
[9]: Row ID          0
      Order ID       0
      Order Date     0
      Date Key       0
      Contact Name    0
      Country         0
      City            0
      Region         0
      Subregion       0
      Customer        0
      Customer ID     0
```



```

Industry      0
Segment       0
Product       0
License       0
Sales         0
Quantity      0
Discount      0
Profit        0
dtype: int64

```

Setelah melakukan penghapusan data anomali, kami melakukan pemeriksaan ulang dan hasilnya menunjukkan bahwa data anomali telah berhasil dihilangkan.

### 2.2.3 Data Filtering

Sebelum kita melanjutkan dengan proses penyaringan data, langkah pertama adalah untuk memahami nilai-nilai unik yang terdapat dalam setiap kolom dataset. Hal ini penting untuk mengidentifikasi nilai-nilai unik yang ada dalam data dan apakah nilai-nilai tersebut sudah tercermin atau mewakili informasi yang sama dalam kolom lain.

```

[10]: # Check Unique Value
display(AWS['Order ID'].value_counts()) # Order ID
display('')
display(AWS['Order Date'].value_counts()) # Order Date
display('')
display(AWS['Date Key'].value_counts()) #Date Key
display('')
display(AWS['Contact Name'].value_counts()) # Contact Name
display('')
display(AWS['Country'].value_counts()) # Country
display('')
display(AWS['City'].value_counts()) # City
display('')
display(AWS['Region'].value_counts()) # Region
display('')
display(AWS['Subregion'].value_counts()) # Sub Region
display('')
display(AWS['Customer'].value_counts()) # Customer
display('')
display(AWS['Customer ID'].value_counts()) # Customer ID
display('')
display(AWS['Industry'].value_counts()) # Industry
display('')
display(AWS['Segment'].value_counts()) # Segment
display('')
display(AWS['Product'].value_counts()) # Product
display('')
display(AWS['License'].value_counts()) # Product License

```

```
display('')
display(AWS['Discount'].value_counts()) # Industry
display('')
```

```
EMEA-2023-100111    14
EMEA-2023-157987    12
AMER-2022-165330    11
EMEA-2022-108504    11
EMEA-2021-131338    10
..
EMEA-2022-101448     1
AMER-2023-117331     1
APJ-2020-127383      1
APJ-2020-110219      1
AMER-2023-119914      1
Name: Order ID, Length: 4999, dtype: int64
```

''

```
9/6/2022           38
9/3/2023           36
11/11/2022         35
12/2/2023          34
12/3/2023          34
..
2/25/2022          1
10/26/2022         1
2/22/2020          1
10/19/2022         1
1/22/2020          1
Name: Order Date, Length: 1236, dtype: int64
```

''

```
20220906           38
20230903           36
20221111           35
20231202           34
20231203           34
..
20220225           1
20221026           1
20200222           1
20221019           1
20200122           1
Name: Date Key, Length: 1236, dtype: int64
```

''

```
Leonard Kelly      37
```

Amelia Campbell	34
Gordon Parr	34
Tracey Dickens	34
Irene Walsh	32
..	
Joshua Gibson	1
Brian Campbell	1
Sam Randall	1
Alexander Duncan	1
Frank Ferguson	1

Name: Contact Name, Length: 793, dtype: int64

''

United States	1996
United Kingdom	1134
Japan	985
France	587
Canada	501
Australia	492
Mexico	469
Germany	383
Brazil	254
Sweden	249
Chile	224
Spain	221
Finland	184
South Africa	183
Russia	182
India	148
South Korea	135
Argentina	129
Ireland	125
Belgium	124
Italy	110
China	104
Philippines	96
Luxembourg	88
Saudi Arabia	81
Poland	66
Singapore	66
Portugal	60
United Arab Emirates	60
Norway	56
Turkey	53
Ukraine	52
Netherlands	45
Czech Republic	42
Colombia	42

Israel	38
Costa Rica	38
New Zealand	37
Greece	27
Indonesia	24
Taiwan	21
Croatia	12
Egypt	11
Austria	10
Denmark	8
Qatar	7
Iceland	4
Slovenia	1

Name: Country, dtype: int64

''

London	915
New York City	738
Paris	537
Los Angeles	509
Toronto	426

'''

Tulsa	1
Chengdu	1
Ljubljana	1
Clinton	1
Tucson	1

Name: City, Length: 262, dtype: int64

''

EMEA	4203
AMER	3653
APJ	2108

Name: Region, dtype: int64

''

NAMER	2497
EU-WEST	1609
UKIR	1259
LATAM	1156
JAPN	985
ANZ	529
NOR	489
APAC	446
MEA	395
EU-EAST	341
IND	148
EU	110

Name: Subregion, dtype: int64

''

Allianz	192
Tyson Foods	184
Ford Motor	175
Siemens	169
Comcast	161

...

Tesco	48
Alphabet	44
Berkshire Hathaway	39
Safeway	39
Citigroup	22

Name: Customer, Length: 99, dtype: int64

''

1080	192
1088	184
1009	175
1043	169
1022	161

...

1059	48
1053	44
1012	39
1079	39
1020	22

Name: Customer ID, Length: 99, dtype: int64

''

Finance	2121
Energy	1312
Tech	1234
Manufacturing	1216
Healthcare	1047
Consumer Products	1018
Retail	969
Communications	592
Transportation	349
Misc	106

Name: Industry, dtype: int64

''

SMB	5178
Strategic	3011
Enterprise	1775

Name: Segment, dtype: int64

```

''

ContactMatcher          1842
Support                 1560
FinanceHub              981
SaaS Connector Pack     957
Site Analytics          887
Marketing Suite - Gold  823
ChatBot Plugin          796
Data Smasher            775
OneView                 466
SaaS Connector Pack - Gold 254
Marketing Suite         228
Storage                 212
Big Ol Database         115
Alchemy                  68
Name: Product, dtype: int64

''

16GRM07R1K      1
Y8TULY7XSD      1
OL8LUB2422      1
4YLZ318U2E      1
NVAYXZ2NB5      1
..
R2G2Z3TP04      1
415425LE8Q      1
AFYQ2WGWOR      1
TC303Z9JVM      1
3NTBV9B3PS      1
Name: License, Length: 9964, dtype: int64

''

0.00    4768
0.20    3657
0.70     418
0.80     300
0.30     227
0.40     206
0.60     138
0.10      94
0.50      66
0.15      52
0.32      27
0.45      11
Name: Discount, dtype: int64

''

```

```
[11]: AWS.drop(['Date Key', 'License', 'Subregion', 'Region', 'Customer ID'], axis = 1,
↳ inplace = True)
```

Setelah kita menganalisis nilai-nilai unik di setiap kolom, kita dapat membuat beberapa kesimpulan:

1. **'Date Key' akan dihapus:** Kolom 'Date Key' akan dihapus karena nilainya hanya hasil konversi dari 'Order Date'.
2. **'Licensed' akan dihapus:** Kolom 'Licensed' akan dihapus karena hanya mencerminkan data lisensi produk.
3. **'Region' dan 'Subregion' akan dihapus:** Kita akan menghapus kolom 'Region' dan 'Subregion' karena kita hanya akan menggunakan data dari kolom 'Country' dan 'City'.
4. **'Customer ID' akan dihapus:** Kolom 'Customer ID' akan dihapus karena informasi ini sudah terwakili dengan baik oleh data 'Customer'.

## 2.2.4 DATA TYPE CONVERTING

Pada langkah ini, kita akan mengubah tipe data kolom 'Order Date' dari objek menjadi tanggal dan waktu (Date Time).

```
[12]: ## Convert Data Order Date From Object to Date Time

AWS['Order Date'] = pd.to_datetime(AWS['Order Date'])
```

Kemudian, kita akan menambahkan tiga kolom baru, yaitu 'Order Month' 'Order Month Name' dan 'Order Year', karena kita akan melakukan analisis data berdasarkan periode bulanan dan tahunan.

```
[13]: ## Add Month
AWS['Order Month'] = AWS['Order Date'].dt.month
## Add Month Name
AWS['Order Month Name']=AWS['Order Date'].dt.month_name()

## Add Year
AWS['Order Year']= AWS['Order Date'].apply (lambda x : str (x)[:4])

AWS.info()
AWS
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9964 entries, 0 to 9993
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Row ID                9964 non-null   int64
1   Order ID              9964 non-null   object
2   Order Date            9964 non-null   datetime64[ns]
3   Contact Name          9964 non-null   object
4   Country               9964 non-null   object
```

```

5   City                9964 non-null  object
6   Customer            9964 non-null  object
7   Industry            9964 non-null  object
8   Segment             9964 non-null  object
9   Product             9964 non-null  object
10  Sales               9964 non-null  float64
11  Quantity            9964 non-null  int64
12  Discount            9964 non-null  float64
13  Profit              9964 non-null  float64
14  Order Month         9964 non-null  int64
15  Order Month Name    9964 non-null  object
16  Order Year          9964 non-null  object
dtypes: datetime64[ns](1), float64(3), int64(3), object(10)
memory usage: 1.4+ MB

```

```

[13]:
      Row ID      Order ID Order Date      Contact Name      Country \
0         1  EMEA-2022-152156 2022-11-09      Nathan Bell      Ireland
1         2  EMEA-2022-152156 2022-11-09      Nathan Bell      Ireland
2         3  AMER-2022-138688 2022-06-13  Deirdre Bailey  United States
3         4  EMEA-2021-108966 2021-10-11      Zoe Hodges      Germany
4         5  EMEA-2021-108966 2021-10-11      Zoe Hodges      Germany
...
9989    9990  EMEA-2020-110422 2020-01-22      Dan Allan      Germany
9990    9991  AMER-2023-121258 2023-02-27  Anthony Sanderson  United States
9991    9992  AMER-2023-121258 2023-02-27  Anthony Sanderson  United States
9992    9993  AMER-2023-121258 2023-02-27  Anthony Sanderson  United States
9993    9994  AMER-2023-119914 2023-05-05  Jasmine Springer  United States

```

```

      City      Customer      Industry      Segment \
0      Dublin      Chevron      Energy      SMB
1      Dublin      Chevron      Energy      SMB
2  New York City  Phillips 66      Energy  Strategic
3      Stuttgart  Royal Dutch Shell      Energy      SMB
4      Stuttgart  Royal Dutch Shell      Energy      SMB
...
9989    Hamburg  Johnson & Johnson      Healthcare      SMB
9990    Milwaukee  American Express      Finance      SMB
9991    Milwaukee  American Express      Finance      SMB
9992    Milwaukee  American Express      Finance      SMB
9993      Dallas      Comcast  Communications      SMB

```

```

      Product      Sales  Quantity  Discount      Profit \
0  Marketing Suite  261.9600         2      0.00    41.9136
1      FinanceHub  731.9400         3      0.00   219.5820
2      FinanceHub   14.6200         2      0.00     6.8714
3  ContactMatcher  957.5775         5      0.45  -383.0310
4  Marketing Suite - Gold  22.3680         2      0.20     2.5164

```



...	...	...	...	...	...
9989	SaaS Connector Pack	25.2480	3	0.20	4.1028
9990	SaaS Connector Pack	91.9600	2	0.00	15.6332
9991	Site Analytics	258.5760	2	0.20	19.3932
9992	Support	29.6000	4	0.00	13.3200
9993	OneView	243.1600	2	0.00	72.9480

	Order Month	Order Month Name	Order Year
0	11	November	2022
1	11	November	2022
2	6	June	2022
3	10	October	2021
4	10	October	2021
...	...	...	...
9989	1	January	2020
9990	2	February	2023
9991	2	February	2023
9992	2	February	2023
9993	5	May	2023

[9964 rows x 17 columns]

Di sini, kita telah mengubah tipe data **'Order Date'** dari awalnya berupa **objek** menjadi tipe data **datetime**. Selain itu, kita telah menambahkan kolom **'Order Month'** dan **'Order Year'**, yang akan berguna untuk analisis.

## 2.3 Checking Statistic Data

### 2.3.1 Data Skewed

Dalam langkah ini, kita akan melakukan analisis distribusi data untuk variabel **Profit**, hasil analisis ini akan menentukan apakah kita akan menggunakan nilai median atau mean serta kita akan cek outlier dari data tersebut.

```
[14]: # Skewed Profit-Terindikasi Right Skewed (Median)
sns.boxplot(AWS['Profit'])
plt.title('Distribusi Data Profit')

skewness = stats.skew(AWS['Profit'])

if skewness > 0:
    print("Data cenderung condong ke kanan (positif skew)")
elif skewness < 0:
    print("Data cenderung condong ke kiri (negatif skew)")
else:
    print("Data memiliki distribusi simetris (tidak ada skew)")

# Mengidentifikasi outliers terbesar dan terkecil
```

```

Q1 = AWS['Profit'].quantile(0.25)
Q3 = AWS['Profit'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers_low = AWS['Profit'][AWS['Profit'] < lower_bound]
outliers_high = AWS['Profit'][AWS['Profit'] > upper_bound]

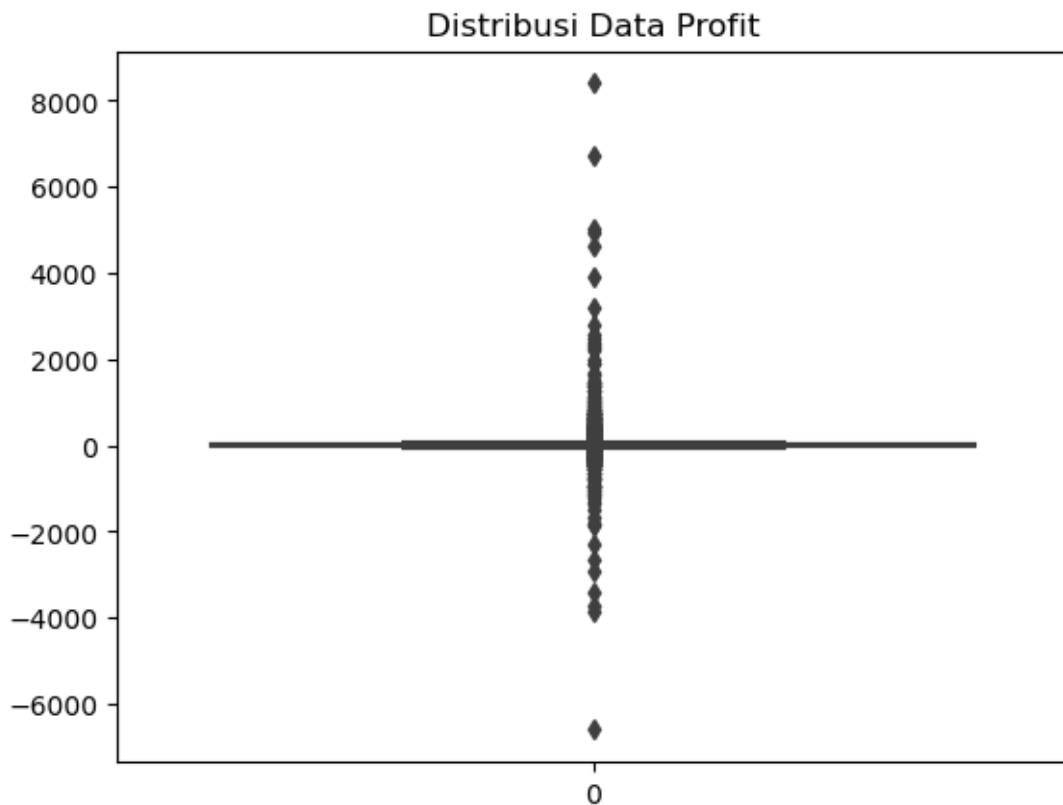
print("Outliers terkecil:")
print(outliers_low.min())

print("Outliers terbesar:")
print(outliers_high.max())

plt.show()

```

Data cenderung condong ke kanan (positif skew)  
 Outliers terkecil:  
 -6599.978  
 Outliers terbesar:  
 8399.976



Hasil dari analisis ini mengindikasikan bahwa distribusi data **Profit** cenderung “**Right Skewed**” (memuai ke kanan), yang berarti kita akan menggunakan nilai median sebagai ukuran tendensi sentral yang lebih sesuai.

Untuk Outliers Terbesar **Profit 8399.976**, Untuk Outlier yang paling kecil nya **-6599.978**

Dalam kasus ini outlier masih dalam batas wajar mengingat bisnis model AWS secara B2B

---

Dalam langkah ini, kita akan melakukan analisis distribusi data untuk variabel **Sales**, hasil analisis ini akan menentukan apakah kita akan menggunakan nilai median atau mean.

```
[15]: # Menggunakan sns.boxplot untuk menampilkan box plot data Sales
sns.boxplot(AWS['Sales'])
plt.title('Distribusi Data Sales')

# Menghitung skewness dari data Sales
skewness = stats.skew(AWS['Sales'])

if skewness > 0:
    print("Data cenderung condong ke kanan (positif skew)")
elif skewness < 0:
    print("Data cenderung condong ke kiri (negatif skew)")
else:
    print("Data memiliki distribusi simetris (tidak ada skew)")

# Mengidentifikasi outliers terbesar dan terkecil
Q1 = AWS['Sales'].quantile(0.25)
Q3 = AWS['Sales'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

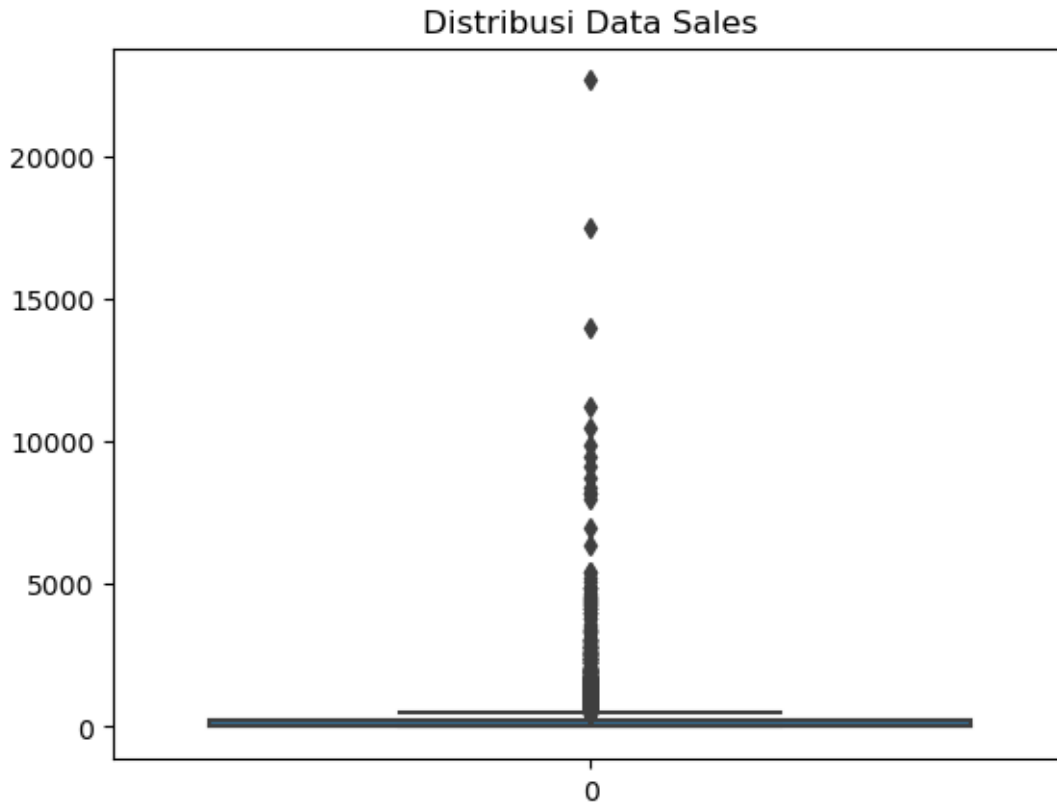
outliers_low = AWS['Sales'][AWS['Sales'] < lower_bound]
outliers_high = AWS['Sales'][AWS['Sales'] > upper_bound]

# Cetak outliers terkecil dan terbesar
print("Outlier terkecil:", outliers_low.min())
print("Outlier terbesar:", outliers_high.max())
```

Data cenderung condong ke kanan (positif skew)

Outlier terkecil: nan

Outlier terbesar: 22638.48



Hasil dari analisis ini mengindikasikan bahwa distribusi data **Sales** cenderung “**Right Skewed**” (memuai ke kanan), yang berarti kita akan menggunakan nilai **median** sebagai ukuran tendensi sentral yang lebih sesuai.

Untuk Outliers Terbesar **Sales 22638**

Dalam kasus ini outlier masih dalam batas wajar mengingat bisnis model AWS secara B2B

Dalam langkah ini, kita akan melakukan analisis distribusi data untuk variabel **Quantity**, hasil analisis ini akan menentukan apakah kita akan menggunakan nilai median atau mean.

```
[16]: # Menggunakan sns.distplot untuk menampilkan distribusi data Quantity
sns.distplot(AWS['Quantity'])
plt.title('Distribusi Data Quantity')

# Menghitung skewness dari data Quantity
skewness = stats.skew(AWS['Quantity'])

if skewness > 0:
    print("Data cenderung condong ke kanan (positif skew)")
elif skewness < 0:
```

```

    print("Data cenderung condong ke kiri (negatif skew)")
else:
    print("Data memiliki distribusi simetris (tidak ada skew)")

# Mengidentifikasi outliers terbesar dan terkecil
Q1 = AWS['Quantity'].quantile(0.25)
Q3 = AWS['Quantity'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers_low = AWS['Quantity'][AWS['Quantity'] < lower_bound]
outliers_high = AWS['Quantity'][AWS['Quantity'] > upper_bound]

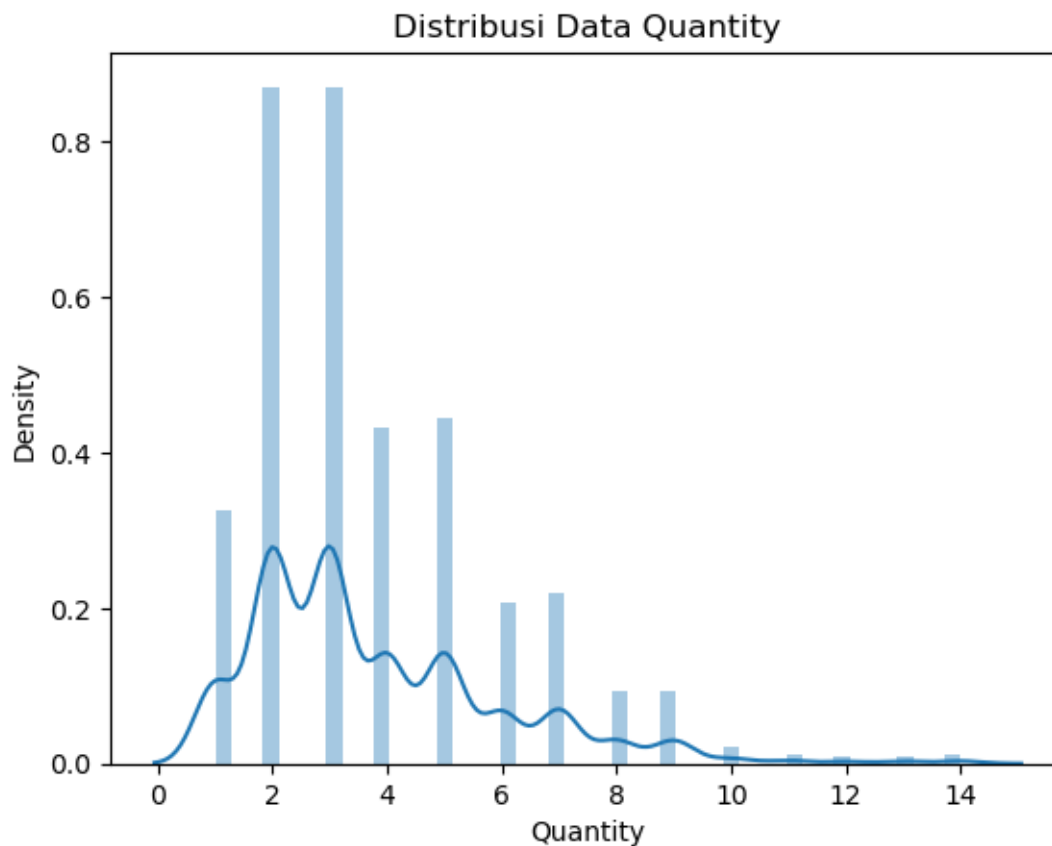
# Cetak outliers terkecil dan terbesar
print("Outlier terkecil:", outliers_low.min())
print("Outlier terbesar:", outliers_high.max())

```

Data cenderung condong ke kanan (positif skew)

Outlier terkecil: nan

Outlier terbesar: 14



Hasil dari analisis ini mengindikasikan bahwa distribusi data **Quantity** cenderung “**Right Skewed**” (memuai ke kanan), yang berarti kita akan menggunakan nilai **median** sebagai ukuran tendensi sentral yang lebih sesuai.

Untuk Outliers Terbesar **Quantity 14**

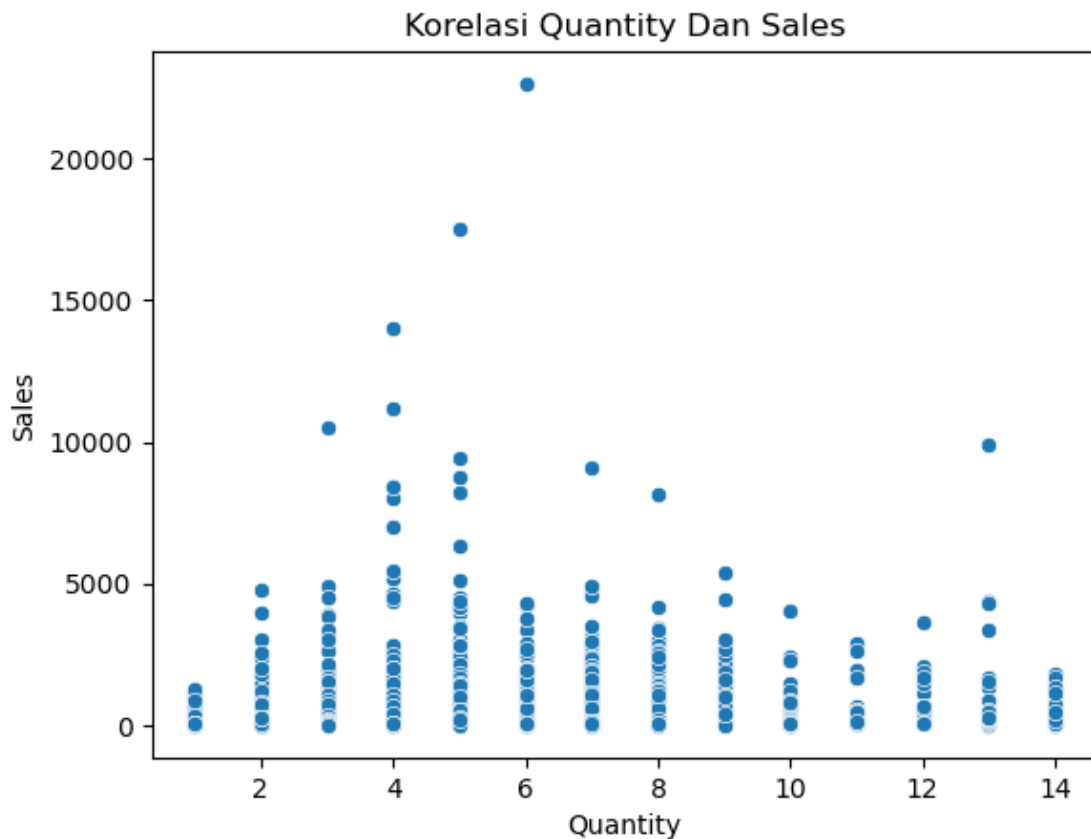
Dalam kasus ini outlier masih dalam batas wajar mengingat bisnis model AWS secara B2B dan Quantity nya masih masuk akal

### 2.3.2 Korelasi

Pada langkah ini, kita akan melakukan uji korelasi antara variabel Quantity dan Sales.

```
[17]: sns.scatterplot(x=AWS['Quantity'],y=AWS['Sales'])  
plt.title('Korelasi Quantity Dan Sales')  
  
spearman_corr, _ = stats.spearmanr(AWS['Quantity'], AWS['Sales'])  
print("Koefisien Korelasi Spearman:", spearman_corr)
```

Koefisien Korelasi Spearman: 0.32684937611373116



Hasil dari uji ini menunjukkan bahwa nilai korelasi Spearman antara keduanya adalah 0.32. Ini menunjukkan adanya korelasi positif antara jumlah produk yang terjual (Quantity) dan penjualan (Sales), namun korelasi tersebut tidak signifikan secara kuat.

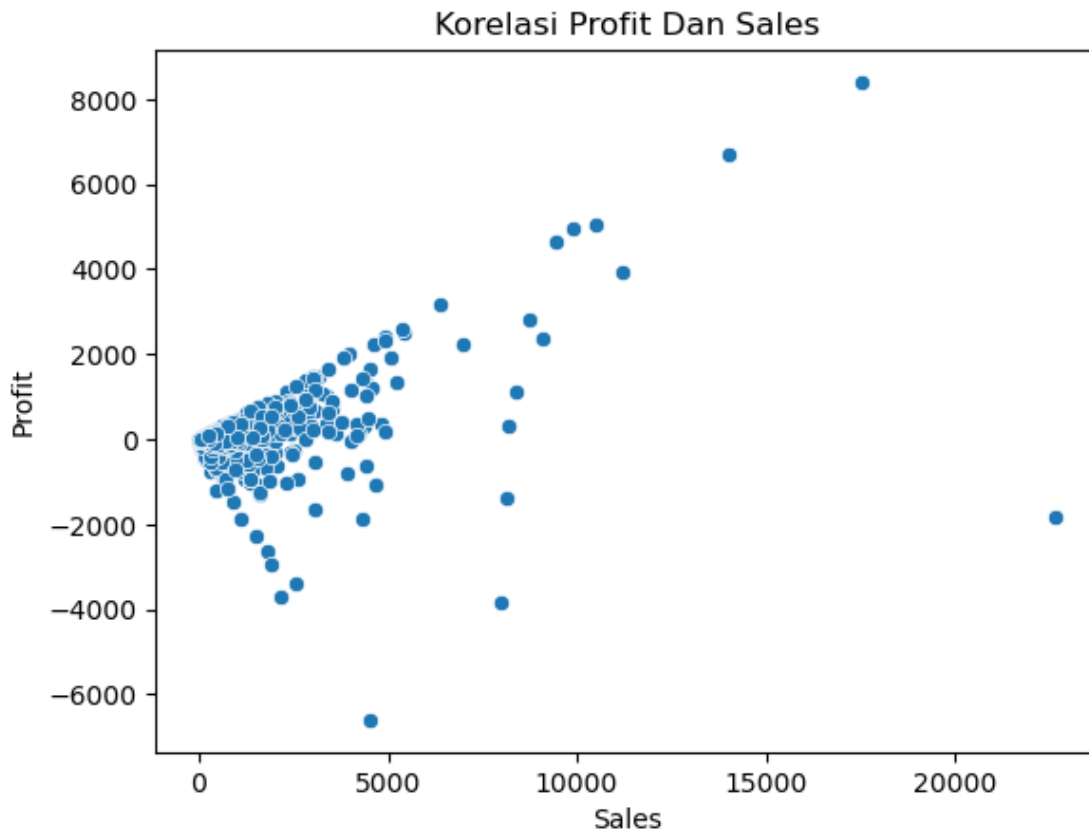
---

Pada tahap ini, kita akan melakukan uji korelasi antara variabel Sales dan Profit.

```
[18]: #Korelasi Profit dan Sales
sns.scatterplot(x=AWS['Sales'],y=AWS['Profit'])
plt.title('Korelasi Profit Dan Sales')

spearman_corr, _ = stats.spearmanr(AWS['Sales'], AWS['Profit'])
print("Koefisien Korelasi Spearman:", spearman_corr)
```

Koefisien Korelasi Spearman: 0.5230857587644749



Hasil dari uji ini menunjukkan bahwa nilai korelasi Spearman antara keduanya adalah 0.52. Hal ini menunjukkan adanya korelasi positif antara penjualan (Sales) dan keuntungan (Profit), meskipun korelasi tersebut tidak signifikan secara kuat.

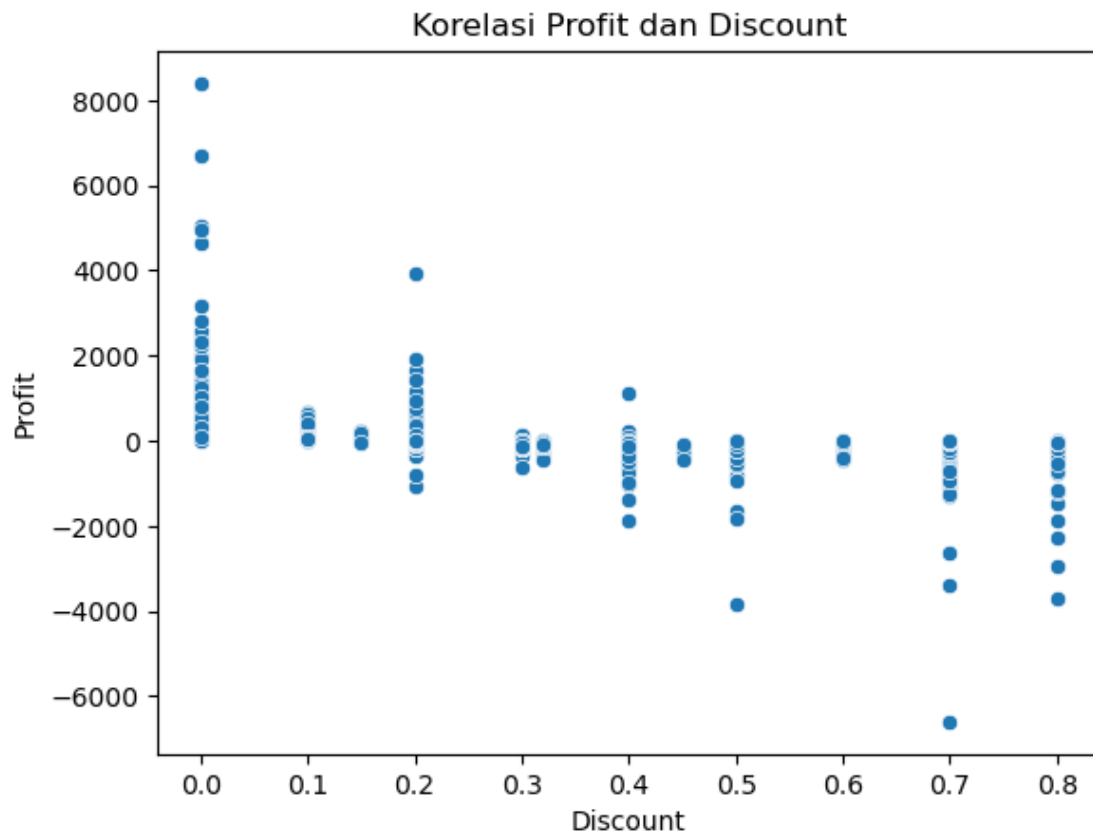
---

Pada langkah ini, kita akan melakukan uji korelasi antara variabel Discount dan Profit.

```
[19]: #Korelasi Profit Dan Discount
sns.scatterplot(x=AWS['Discount'],y=AWS['Profit'])
plt.title('Korelasi Profit dan Discount')

spearman_corr, _ = stats.spearmanr(AWS['Profit'], AWS['Discount'])
print("Koefisien Korelasi Spearman:", spearman_corr)
```

Koefisien Korelasi Spearman: -0.5473580539549545



Hasil dari uji ini menunjukkan bahwa nilai korelasi Spearman antara keduanya adalah -0.54. Ini menunjukkan adanya korelasi negatif antara diskon (Discount) dan keuntungan (Profit), meskipun korelasi tersebut tidak signifikan secara kuat.

### 3 Analisis Data

Di sini, kita akan memulai proses analisis data yang mencakup beberapa aspek penting.



1. **Timeline Sales:** Analisis ini akan memberikan wawasan tentang bagaimana penjualan berfluktuasi sepanjang waktu, termasuk tren peningkatan atau penurunan.
2. **Timeline Profit:** Melalui analisis ini, kita dapat memahami bagaimana profit berubah seiring berjalannya waktu dan mengidentifikasi periode dengan kinerja keuntungan yang baik atau buruk.
3. **Discount by Month:** Analisis diskon per bulan akan membantu kita melihat pola penggunaan diskon dan apakah ada bulan-bulan tertentu yang lebih efektif dalam meningkatkan penjualan atau profitabilitas.
4. **Sales by Country:** Dengan menganalisis penjualan berdasarkan negara, kita dapat mengidentifikasi pasar yang paling kuat dan mungkin mengevaluasi strategi ekspansi ke negara-negara baru.
5. **Segment Sales:** Analisis ini akan membantu kita memahami kinerja penjualan di berbagai segmen bisnis, yang dapat membantu dalam penyesuaian strategi pemasaran dan produk.
6. **Company Sales:** Menilai penjualan per perusahaan akan memungkinkan kita untuk mengidentifikasi kontributor utama terhadap pendapatan keseluruhan dan memahami peran masing-masing perusahaan dalam portofolio bisnis.
7. **Product Profit:** Melalui analisis profit produk, kita dapat menilai produk mana yang memberikan profitabilitas tertinggi dan apakah ada produk yang memerlukan peninjauan atau pengembangan lebih lanjut.

Dengan menganalisis semua aspek ini, kita akan memiliki wawasan yang komprehensif tentang kinerja bisnis dan dapat membuat keputusan yang lebih terinformasi untuk pertumbuhan dan keberhasilan bisnis ke depan.

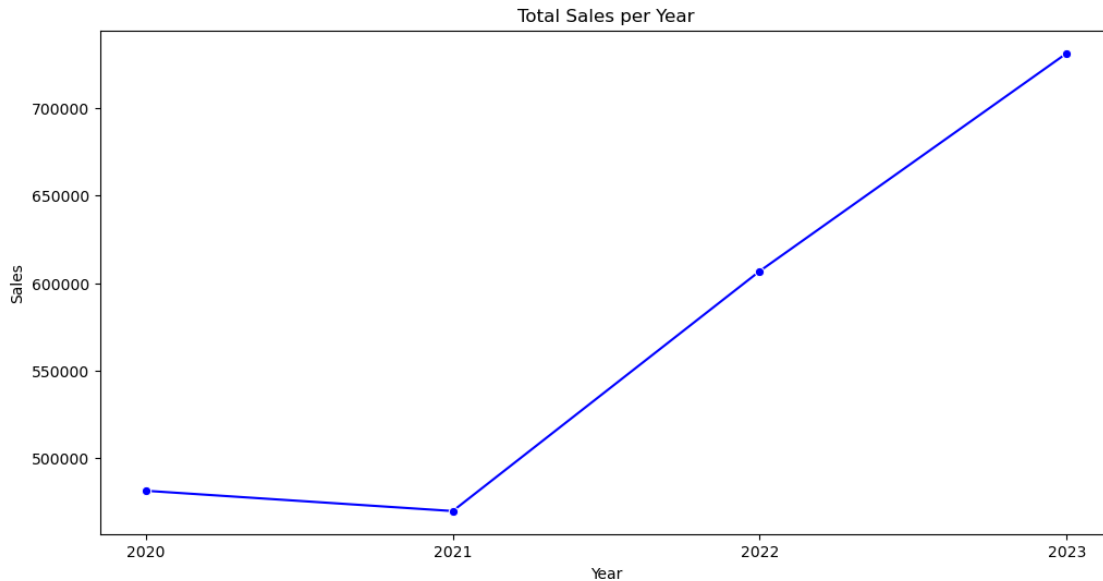
### 3.1 Time by Time Sales

Pada tahap ini kita akan melakukan analisa total penjualan per tahunnya.

```
[20]: yearlySales = AWS.groupby('Order Year')['Sales'].sum().reset_index()

plt.figure(figsize=(12, 6))
sns.lineplot(x='Order Year', y='Sales', data=yearlySales, marker='o',
             color='blue')
plt.title('Total Sales per Year')
plt.xlabel('Year')
plt.ylabel('Sales')
```

```
[20]: Text(0, 0.5, 'Sales')
```



Secara keseluruhan, penjualan setiap tahunnya mengalami pertumbuhan, meskipun terdapat penurunan pada tahun 2021 yang kemungkinan terkait dengan masa pandemi. Namun, penjualan kembali meningkat pada tahun 2022 dan 2023.

Pada tahap ini kita akan melakukan analisis total penjualan per bulan untuk setiap tahunnya.

```
[21]: # Menghitung total Sales per bulan dan tahun
monthlySales = AWS.groupby(['Order Month', 'Order Year'])['Sales'].sum().
    ↪reset_index()

# Mengubah tipe data 'Order Month' menjadi integer
monthlySales['Order Month'] = monthlySales['Order Month'].astype(int)

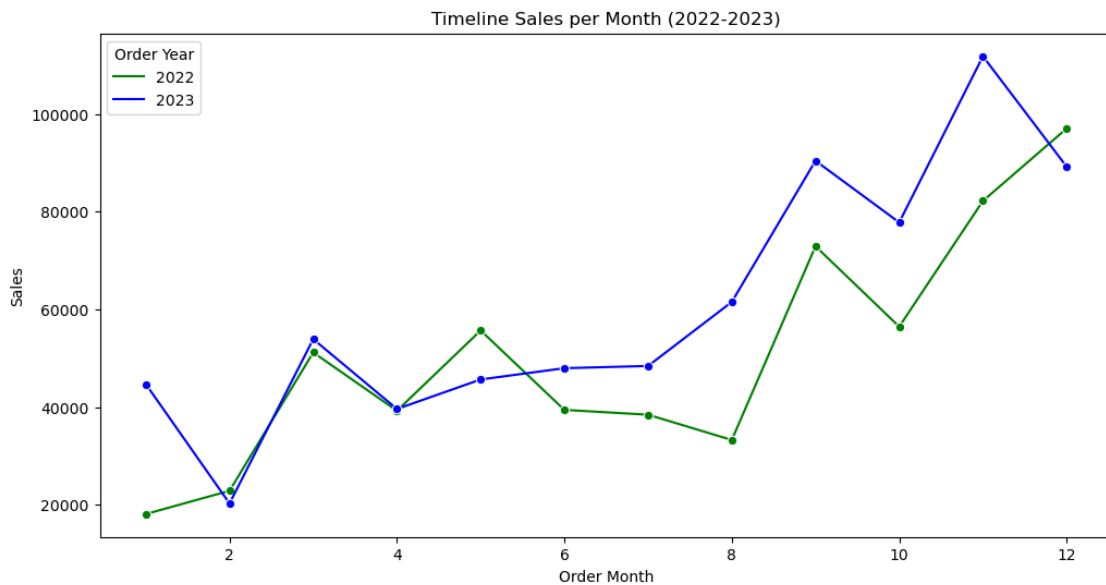
# Filter data untuk tahun 2020 dan 2021
data_2020_2021 = monthlySales[(monthlySales['Order Year'] == '2020') |
    ↪(monthlySales['Order Year'] == '2021')]

# Filter data untuk tahun 2022 dan 2023
data_2022_2023 = monthlySales[(monthlySales['Order Year'] == '2022') |
    ↪(monthlySales['Order Year'] == '2023')]

# Plot data untuk tahun 2020 dan 2021
plt.figure(figsize=(12, 6))
sns.lineplot(x='Order Month', y='Sales', hue='Order Year', data=data_2020_2021,
    ↪marker='o', palette=['red', 'orange'])
plt.title('Timeline Sales per Month (2020-2021)')
plt.ylabel('Sales')
```

```
# Plot data untuk tahun 2022 dan 2023
plt.figure(figsize=(12, 6))
sns.lineplot(x='Order Month', y='Sales', hue='Order Year', data=data_2022_2023,
             marker='o', palette=['green', 'blue'])
plt.title('Timeline Sales per Month (2022-2023)')
plt.ylabel('Sales')

plt.show()
```



Dalam analisis time by time total penjualan bulanan berdasarkan tahun, kita dapat mengamati beberapa hal berikut:

1. Terjadi peningkatan penjualan setiap bulannya, dengan puncak pada bulan November.
2. Februari menunjukkan penjualan terendah setiap tahunnya. Hal ini mungkin disebabkan oleh pembaruan atau peremajaan aset yang dilakukan oleh pelanggan pada akhir tahun, sehingga mereka tidak membutuhkan produk kita dalam periode ini.

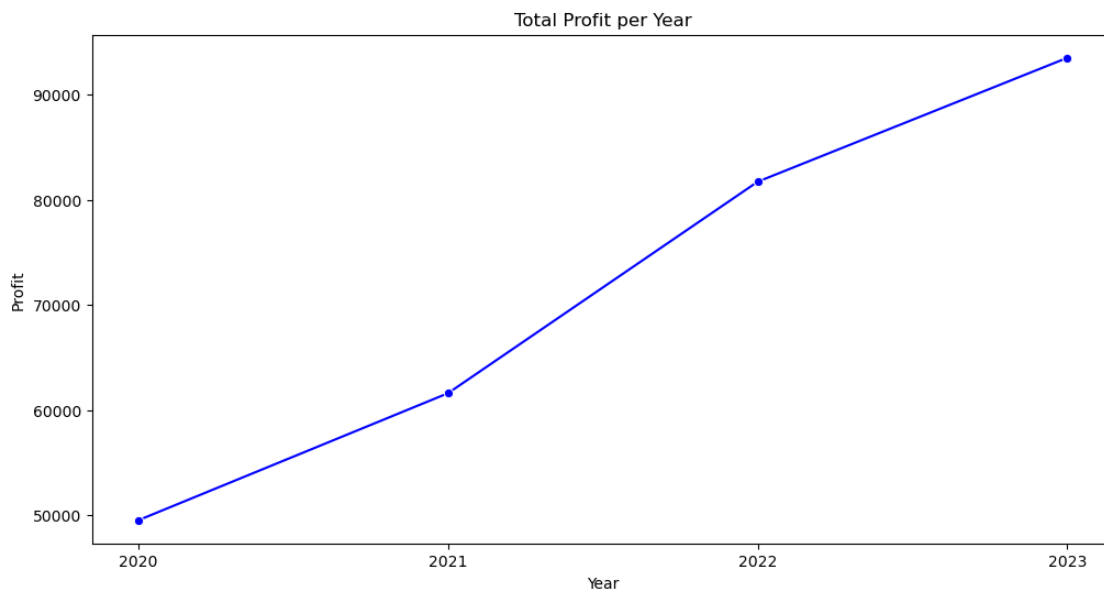
### 3.2 Time by Time Profit

Pada tahap ini kita akan melakukan analisa total profit per tahunnya.

```
[22]: yearlyProfit = AWS.groupby('Order Year')['Profit'].sum().reset_index()

plt.figure(figsize=(12, 6))
sns.lineplot(x='Order Year', y='Profit', data=yearlyProfit, marker='o', color='blue')
plt.title('Total Profit per Year')
plt.xlabel('Year')
plt.ylabel('Profit')
```

```
[22]: Text(0, 0.5, 'Profit')
```



Setelah melakukan analisis terhadap Profit:

1. AWS selalu mengalami kenaikan Profit setiap tahunnya
2. Pada tahun 2021 walaupun AWS mengalami penurunan Penjualan, AWS berhasil meningkatkan nilai Profitnya

---

Pada tahap ini kita akan melakukan analisis total profit per bulan untuk setiap tahunnya.

```
[23]: # Menghitung total Profit per bulan dan tahun
monthlyProfit = AWS.groupby(['Order Month', 'Order Year'])['Profit'].sum().
      ↪reset_index()

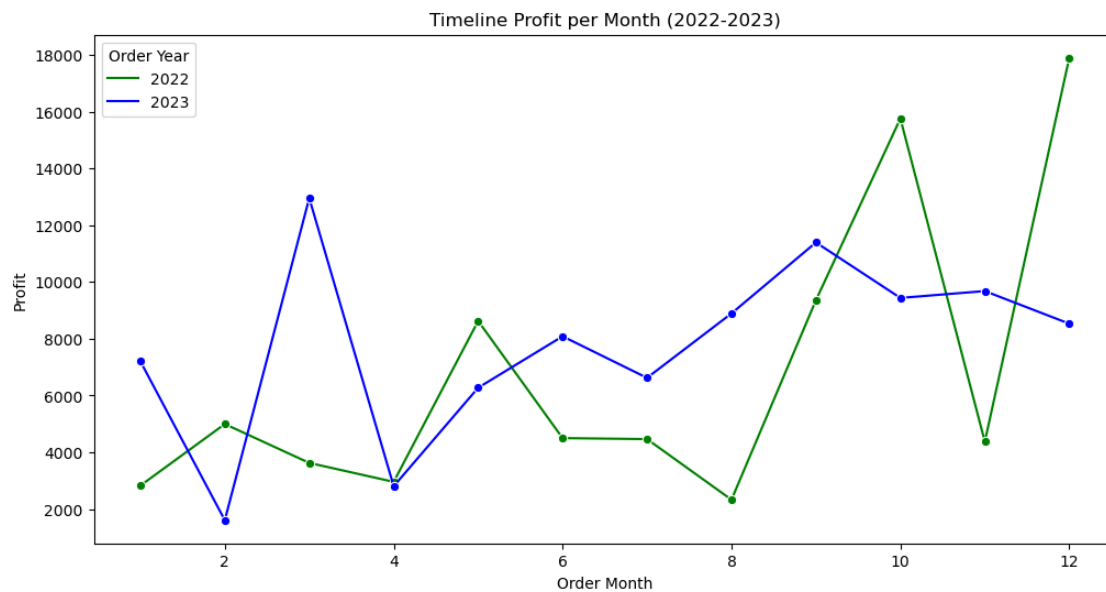
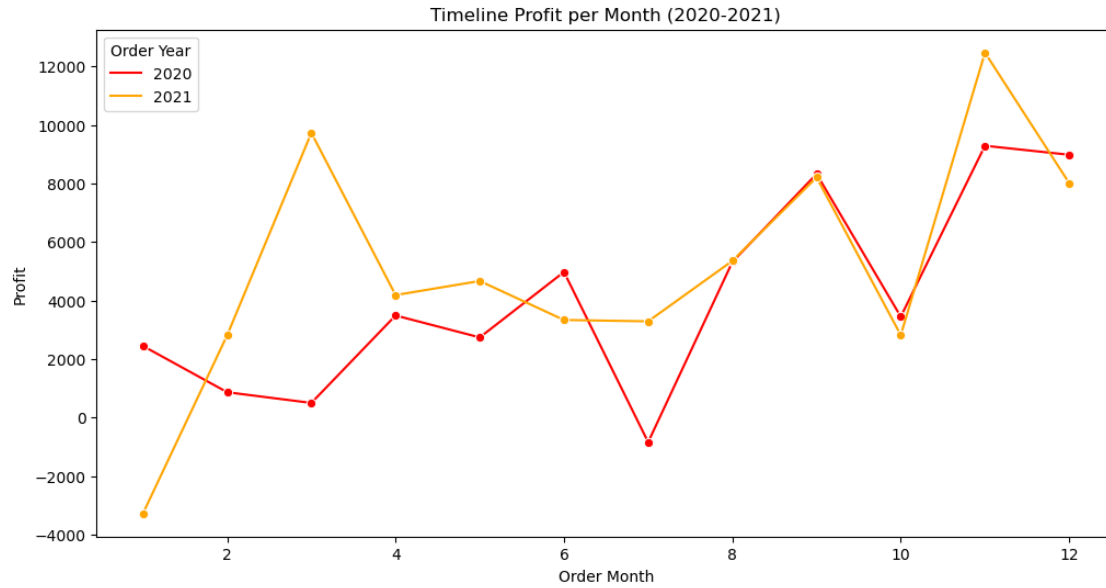
# Filter data untuk tahun 2020 dan 2021
data_2020_2021 = monthlyProfit[(monthlyProfit['Order Year'] == '2020') |
      ↪(monthlyProfit['Order Year'] == '2021')]

# Filter data untuk tahun 2022 dan 2023
data_2022_2023 = monthlyProfit[(monthlyProfit['Order Year'] == '2022') |
      ↪(monthlyProfit['Order Year'] == '2023')]

# Plot data untuk tahun 2020 dan 2021
plt.figure(figsize=(12, 6))
sns.lineplot(x='Order Month', y='Profit', hue='Order Year',
      ↪data=data_2020_2021, marker='o', palette=['red', 'orange'])
plt.title('Timeline Profit per Month (2020-2021)')
plt.ylabel('Profit')

# Plot data untuk tahun 2022 dan 2023
plt.figure(figsize=(12, 6))
sns.lineplot(x='Order Month', y='Profit', hue='Order Year',
      ↪data=data_2022_2023, marker='o', palette=['green', 'blue'])
plt.title('Timeline Profit per Month (2022-2023)')
plt.ylabel('Profit')

plt.show()
```



Dalam analisis time by time total profit bulanan berdasarkan tahun, kita bisa mengamati beberapa temuan penting:

1. Grafik profit masih memiliki karakteristik yang fluktuatif, mungkin karena efek dari kampanye diskon yang belum terukur sepenuhnya.
2. Pada bulan Januari 2021, kita mengalami kerugian yang mungkin dapat diatribusikan ke beberapa faktor yang mempengaruhi kinerja bisnis pada waktu itu.
3. Puncak profit tertinggi terjadi pada tahun 2022, menunjukkan peningkatan yang signifikan

dalam kinerja bisnis pada tahun tersebut.

### 3.3 Discount by Month

Pada tahap ini, kita akan menganalisis jumlah Rata-rata diskon yang diberikan setiap bulannya dari setiap tahunnya.

```
[24]: # Menghitung rata-rata 'Discount' per bulan dan tahun
monthly_avg_discount = AWS.groupby(['Order Month', 'Order Year'])['Discount'].
    ↪mean().reset_index()

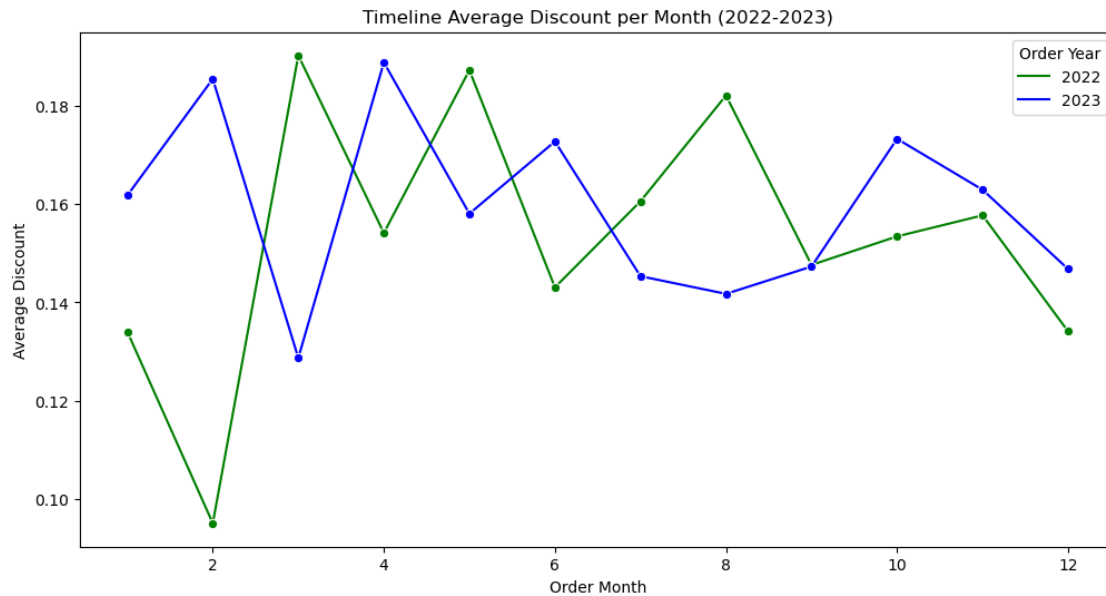
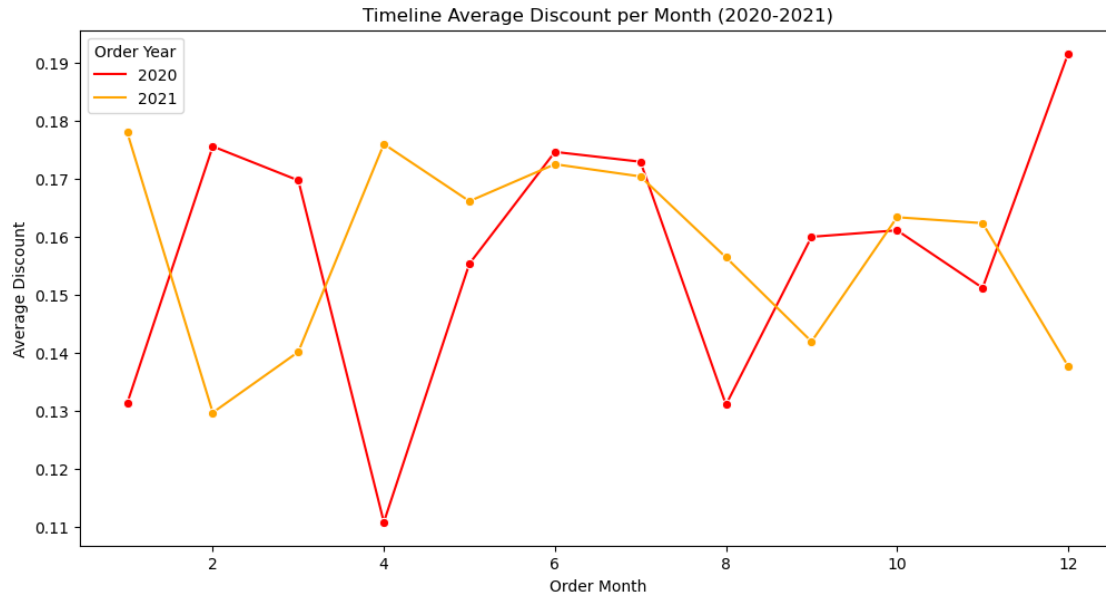
# Filter data untuk tahun 2020 dan 2021
data_2020_2021 = monthly_avg_discount[(monthly_avg_discount['Order Year'] ==
    ↪'2020') | (monthly_avg_discount['Order Year'] == '2021')]

# Filter data untuk tahun 2022 dan 2023
data_2022_2023 = monthly_avg_discount[(monthly_avg_discount['Order Year'] ==
    ↪'2022') | (monthly_avg_discount['Order Year'] == '2023')]

# Plot data untuk tahun 2020 dan 2021
plt.figure(figsize=(12, 6))
sns.lineplot(x='Order Month', y='Discount', hue='Order Year',
    ↪data=data_2020_2021, marker='o', palette=['red', 'orange'])
plt.title('Timeline Average Discount per Month (2020-2021)')
plt.ylabel('Average Discount')

# Plot data untuk tahun 2022 dan 2023
plt.figure(figsize=(12, 6))
sns.lineplot(x='Order Month', y='Discount', hue='Order Year',
    ↪data=data_2022_2023, marker='o', palette=['green', 'blue'])
plt.title('Timeline Average Discount per Month (2022-2023)')
plt.ylabel('Average Discount')

plt.show()
```



Berdasarkan analisis rata-rata discount per bulan, kami melihat bahwa mayoritas, ketika grafik profit mengalami penurunan, grafik rata-rata discount cenderung meningkat, dan sebaliknya. Dari hasil ini, kami dapat menyimpulkan bahwa nilai discount secara mempengaruhi nilai profit.

Oleh karena itu, pengelolaan diskon perlu diperhatikan dengan cermat, dan keputusan terkait diskon harus didasarkan pada analisis data yang cermat untuk memahami dampaknya terhadap profit dan penjualan



## 3.4 Sales By Country

### 3.4.1 Top 5 Sales Country

Disini kita akan analisa Top 5 Country By Sales

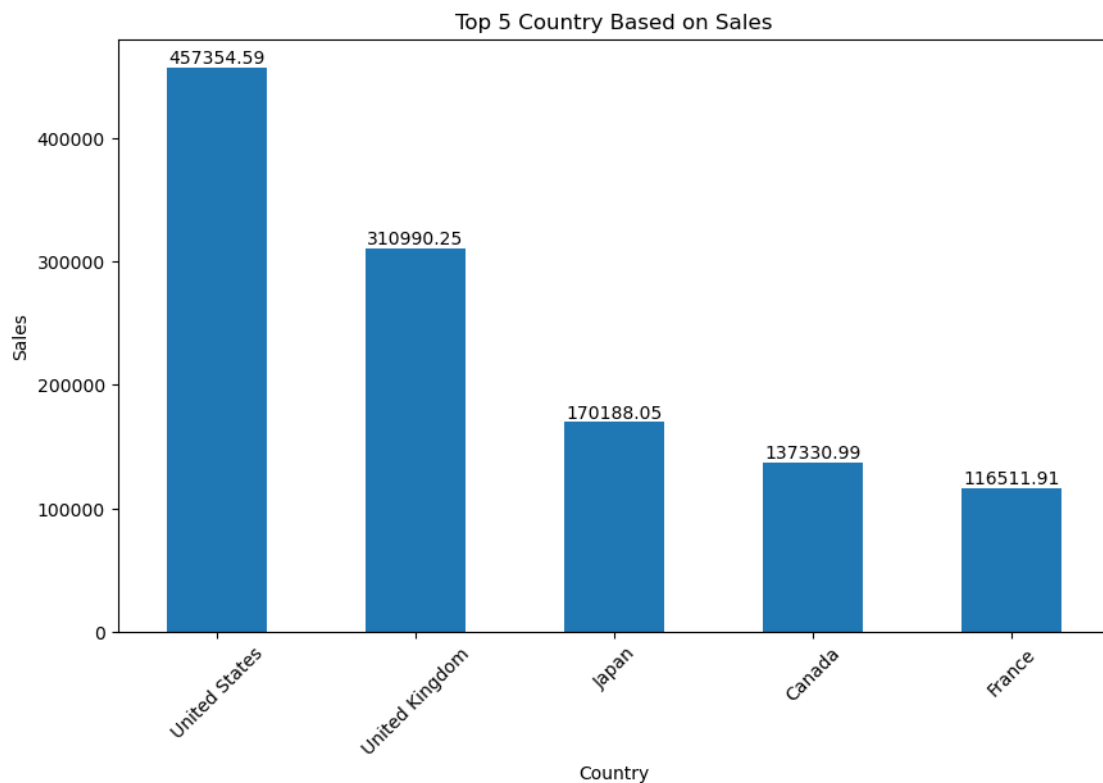
```
[25]: Sales_Country = AWS[['Country', 'Sales']].groupby('Country').sum().
      ↪sort_values(by='Sales', ascending=False).head(5)

# Membuat grafik batang
ax = Sales_Country.plot(kind='bar', legend=False, figsize=(10, 6))

# Menambahkan label pada batang
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.
    ↪get_height()), ha='center', va='bottom')

plt.title('Top 5 Country Based on Sales')
plt.xlabel('Country')
plt.ylabel('Sales')
plt.xticks(rotation=45) # Untuk memutar label negara agar lebih mudah dibaca

plt.show()
```



Dalam analisis top 5 negara berdasarkan penjualan, kita dapat mengamati beberapa temuan yang dapat membantu dalam pengambilan keputusan:

1. Negara teratas dalam penjualan adalah AS (US), diikuti oleh Inggris (UK) sebagai peringkat kedua, dan Jepang sebagai peringkat ketiga.
2. Ada peluang untuk meluncurkan kampanye yang berfokus pada pelanggan setia (Loyal Customer) di negara-negara teratas ini untuk mempertahankan dan meningkatkan penjualan.
3. Selain itu, kampanye yang mengakui pelanggan dengan memberikan apresiasi dan hadiah mungkin dapat memperkuat ikatan dengan pelanggan dan memotivasi mereka untuk terus berbelanja.

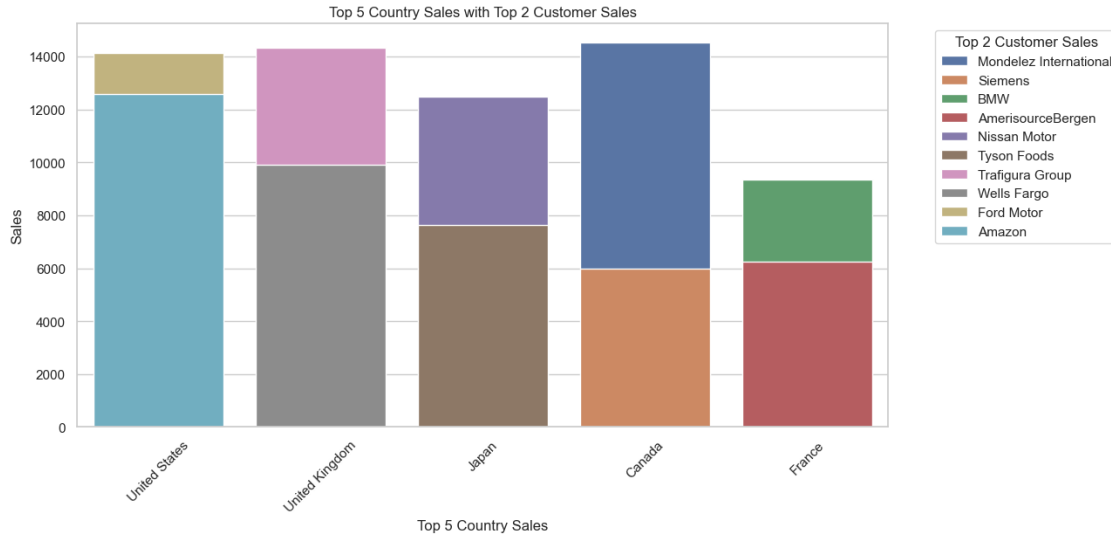
Langkah-langkah ini dapat membantu dalam mengoptimalkan strategi pemasaran dan pertumbuhan bisnis di negara-negara ini.

---

Setelah mengidentifikasi lima negara teratas, kita dapat memilih dua pelanggan terbaik dari masing-masing negara ini untuk menjalankan kampanye yang lebih terfokus dan terukur

```
[26]: country_sales = AWS.groupby('Country')['Sales'].sum().reset_index()
country_sales = country_sales.sort_values(by='Sales', ascending=False)
top_5_countries = country_sales.nlargest(5, 'Sales')
AWS_filtered = AWS[AWS['Country'].isin(top_5_countries['Country'])]
customer_sales = AWS_filtered.groupby(['Country', 'Customer'])['Sales'].sum().
    ↪reset_index()
top_5_customers = customer_sales.groupby('Country').apply(lambda x: x.
    ↪nlargest(2, 'Sales')).reset_index(drop=True)
```

```
[27]: plt.figure(figsize=(12, 6))
sns.set(style="whitegrid", font_scale=1)
sns.barplot(data=top_5_customers, x='Country', y='Sales', hue='Customer',
    ↪ci=0,dodge=False,order=top_5_countries['Country'])
plt.title('Top 5 Country Sales with Top 2 Customer Sales')
plt.xlabel('Top 5 Country Sales')
plt.ylabel('Sales')
plt.xticks(rotation=45)
plt.legend(title='Top 2 Customer Sales', bbox_to_anchor=(1.05, 1), loc='upper_
    ↪left')
plt.show()
```



1. Di AS (US), pelanggan terbaik adalah Amazon yang menduduki peringkat pertama, diikuti oleh Ford Motor sebagai peringkat kedua.
2. Di Inggris (UK), pelanggan terbaik adalah Wells Fargo yang berada di peringkat pertama, dan Trafiguro Group yang menduduki peringkat kedua.
3. Di Jepang, pelanggan terbaik adalah Tyson Food yang menduduki peringkat pertama, diikuti oleh Nissan Motor sebagai peringkat kedua.

Langkah ini akan membantu dalam menjalankan kampanye yang lebih terarah dan berfokus pada pelanggan-pelanggan yang memiliki dampak signifikan pada penjualan di masing-masing negara tersebut.

---

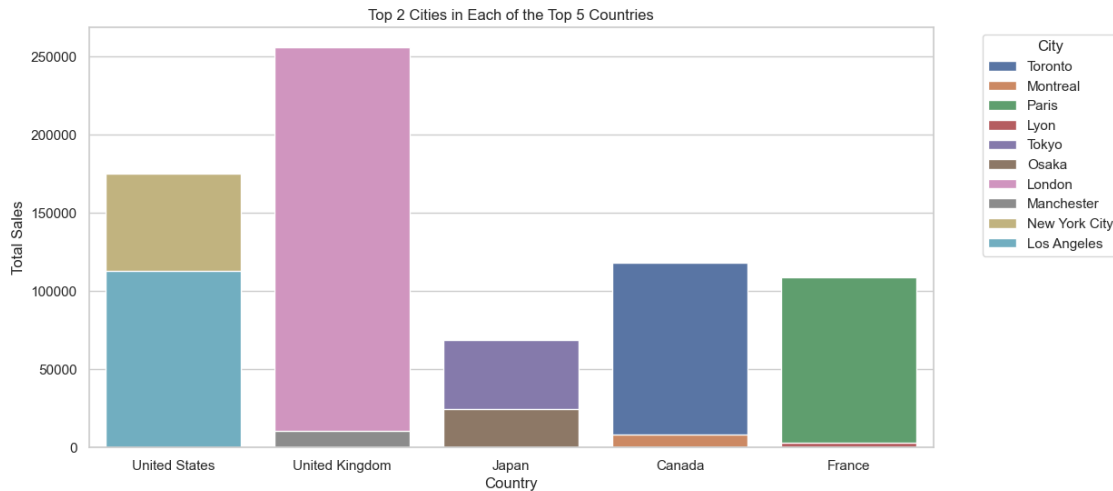
Langkah selanjut nya kita akan mencari Top 2 City dari Top 5 Country nya

```
[28]: city_sales = AWS.groupby('City')['Sales'].sum().reset_index()
country_sales = country_sales.sort_values(by='Sales', ascending=False)
top_5_countries = country_sales.head(5)
filtered_data = AWS[AWS['Country'].isin(top_5_countries['Country'])]

top_2_city_country = filtered_data.groupby(['Country', 'City'])['Sales'].sum().
    ↪reset_index()
top_2_city_country = top_2_city_country.groupby('Country').apply(lambda x: x.
    ↪nlargest(2, 'Sales')).reset_index(drop=True)
```

```
[29]: plt.figure(figsize=(12, 6))
sns.set(style="whitegrid", font_scale=1)
sns.barplot(data=top_2_city_country, x='Country', y='Sales', hue='City',
    ↪estimator=np.sum, ci=0, dodge=False, order=top_5_countries['Country'])
plt.title('Top 2 Cities in Each of the Top 5 Countries')
```

```
plt.xlabel('Country')
plt.ylabel('Total Sales')
plt.legend(title='City',bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



Setelah kita analisa Top 2 kota dari masing-masing lima negara teratas, kita dapat mengidentifikasi:

1. Amerika (US): Toronto dan Montreal sebagai dua kota teratas.
2. Inggris (UK): London dan Manchester sebagai dua kota teratas.
3. Jepang: Tokyo dan Osaka sebagai dua kota teratas.

Dengan langkah ini, kita dapat mengorganisir kampanye yang lebih terukur dan terfokus di kota-kota utama ini, yang memiliki potensi signifikan dalam mendukung pertumbuhan penjualan dan keberhasilan kampanye.

### 3.4.2 Lowest 5 Sales Country

Pada tahap ini kita akan analisa Lowest 5 Sales by Country

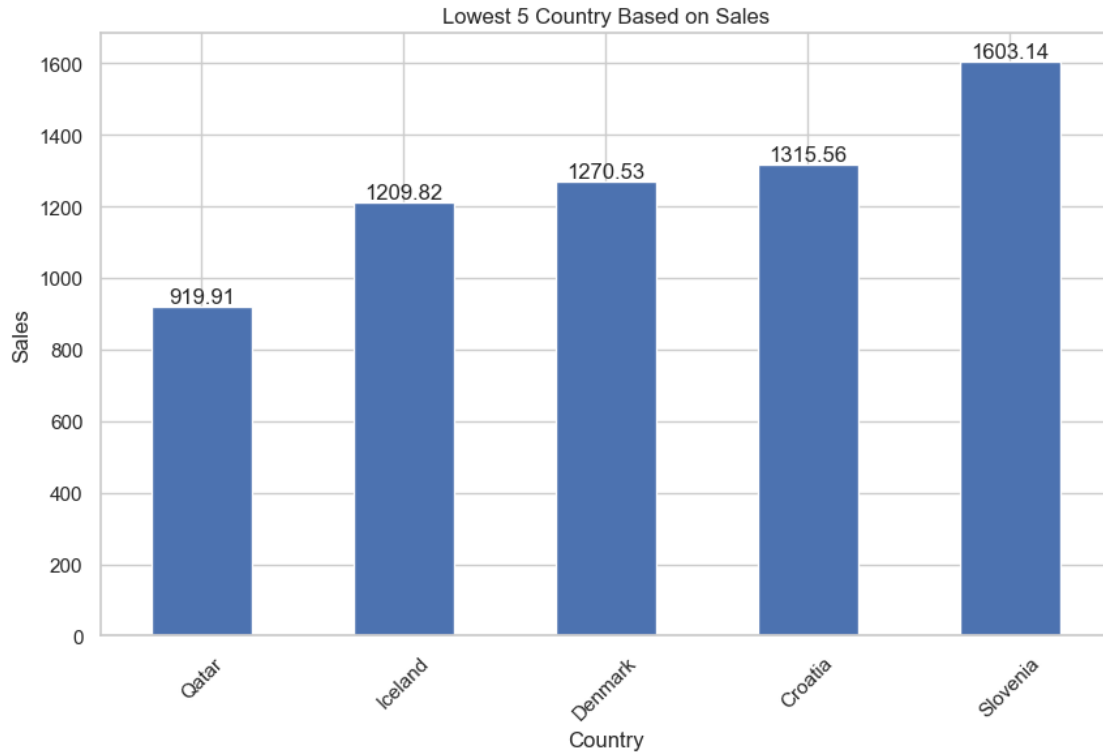
```
[30]: Sales_Country = AWS[['Country', 'Sales']].groupby('Country').sum().
      ↪sort_values(by='Sales', ascending=True).head()

# Membuat grafik batang
ax = Sales_Country.plot(kind='bar', legend=False, figsize=(10, 6))

# Menambahkan label pada batang
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.
    ↪get_height()), ha='center', va='bottom')

plt.title('Lowest 5 Country Based on Sales')
plt.xlabel('Country')
```

```
plt.ylabel('Sales')
plt.xticks(rotation=45) # Untuk memutar label negara agar lebih mudah dibaca
plt.show()
```



Dalam melihat lima negara dengan penjualan terendah, kita mengidentifikasi:

1. Qatar sebagai negara dengan penjualan terendah.
2. Islandia sebagai peringkat kedua
3. Denmark sebagai peringkat ketiga.

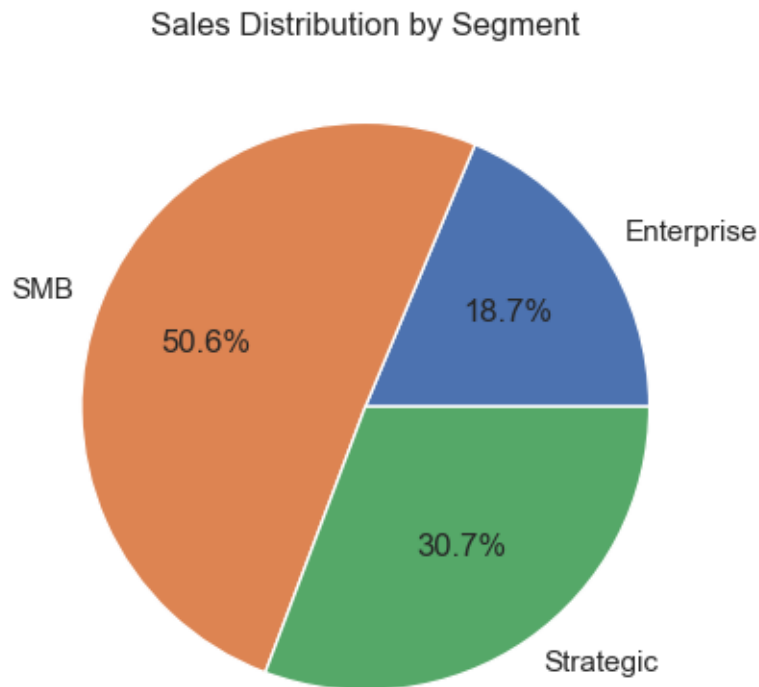
Langkah yang bisa diambil adalah untuk melakukan kampanye yang bertujuan untuk menjajaki potensi “Pasar Baru” di negara-negara ini dan mendorong mereka untuk melakukan transaksi. Dengan cara ini, kita dapat mengidentifikasi peluang pertumbuhan baru di negara-negara dengan penjualan rendah ini.

### 3.5 Segment Sales

Dalam tahap ini, kita menganalisis distribusi penjualan berdasarkan proporsinya:

```
[31]: sns.set(style="whitegrid", font_scale=1)
region_sales = AWS.groupby('Segment')['Sales'].sum()
plt.pie(region_sales, labels=region_sales.index, autopct='%1.1f%%')
plt.title('Sales Distribution by Segment')
```

```
plt.show()
```



Proporsi penjualan adalah sebagai berikut: - SMB (Small and Medium-sized Business): 50.6% - Strategic: 30.7% - Enterprise: 18.7%

Ini menggambarkan sejauh mana setiap segmen berkontribusi pada total penjualan, dengan SMB memiliki andil terbesar diikuti oleh segmen Strategic dan Enterprise.

---

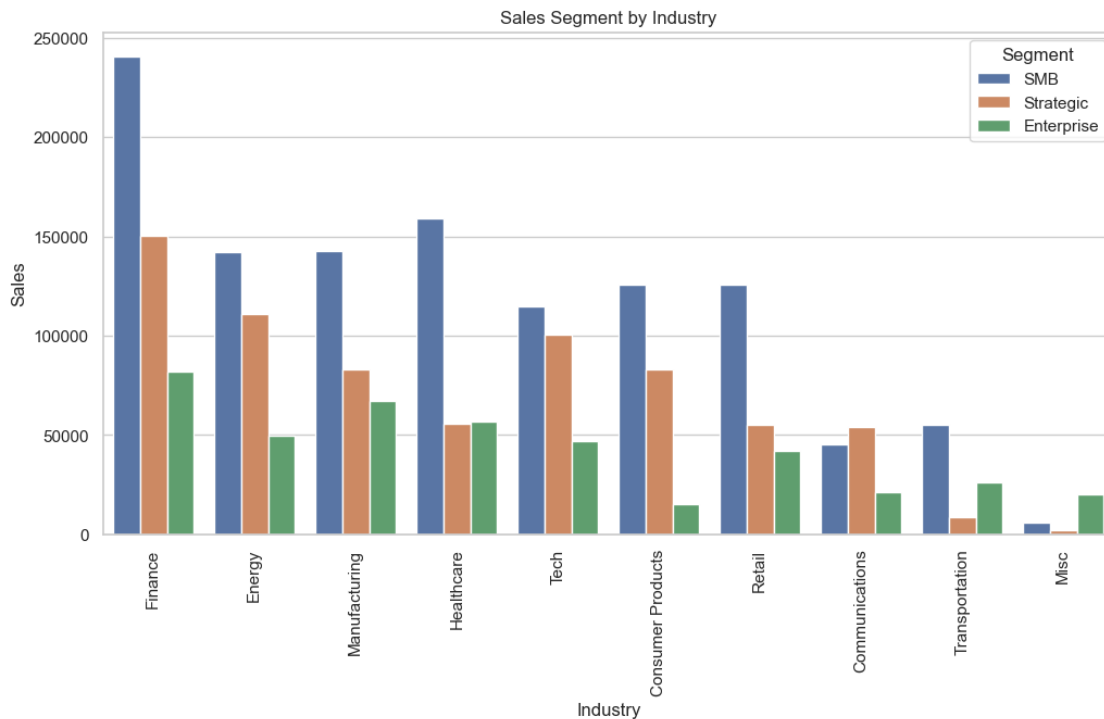
Pada tahap ini, kita menganalisis data penjualan berdasarkan industri dan membaginya berdasarkan segmen.

```
[32]: industry_sales = AWS.groupby('Industry')['Sales'].sum().reset_index().
      ↪sort_values(by='Sales', ascending=False)

plt.figure(figsize=(12,6))
sns.set(style="whitegrid", font_scale=1)
sns.barplot(data=AWS,x='Industry',
            ↪y='Sales',estimator=sum,hue='Segment',ci=0,order=industry_sales['Industry'])

plt.title('Sales Segment by Industry')
plt.xticks(rotation=90)
```

```
plt.show()
```



Dengan data ini, kita dapat:

- Mengidentifikasi fokus kita dalam pengembangan segmen yang sesuai dengan kebutuhan industri tertentu.
- Memahami prioritas pemasaran segmen berdasarkan industri, yang memungkinkan kita untuk menyusun strategi pemasaran yang lebih tepat sasaran.
- Mengevaluasi kinerja penjualan segmen berdasarkan industri, sehingga kita dapat mengalokasikan sumber daya dengan lebih efisien untuk pertumbuhan bisnis yang berkelanjutan.

### 3.6 Customer/Company Sales

Pada tahap ini, kita akan mengumpulkan data mengenai perusahaan-perusahaan teratas berdasarkan penjualan.

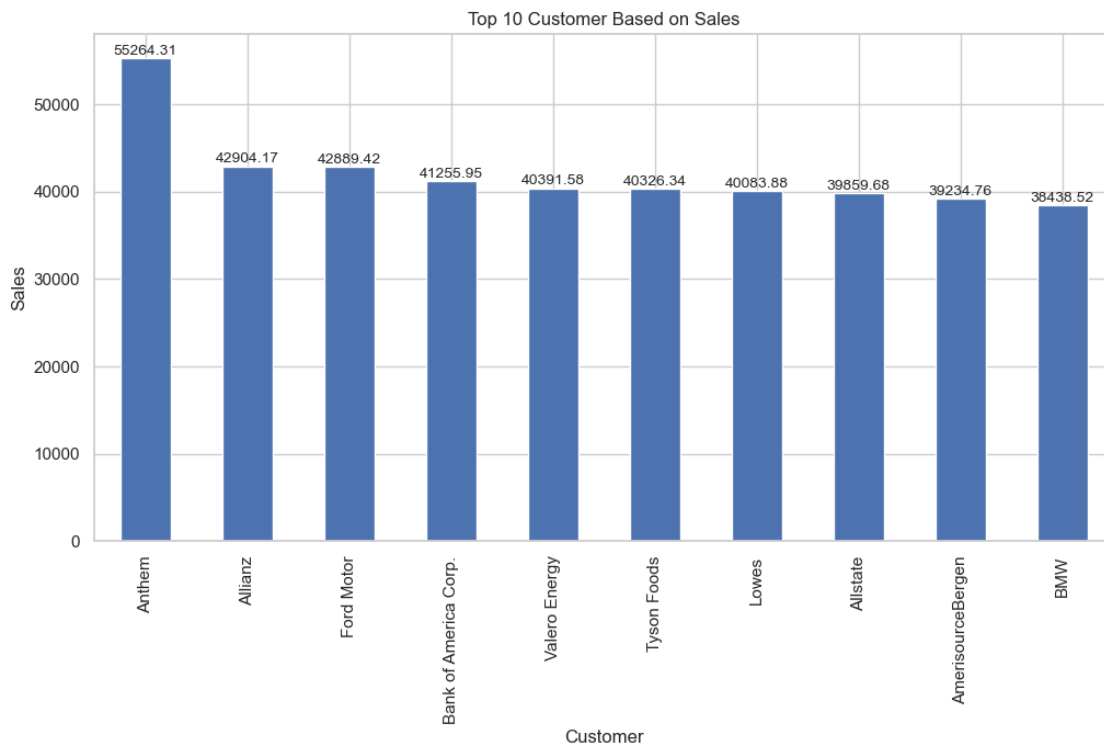
```
[33]: Sales_Customer = AWS[['Customer', 'Sales']].groupby('Customer').sum().
      ↪sort_values(by='Sales', ascending=False).head(10)

# Membuat grafik batang
ax = Sales_Customer.plot(kind='bar', legend=False, figsize=(12, 6))

# Menambahkan label pada batang
for i, v in enumerate(Sales_Customer['Sales']):
    ax.text(i, v + 50, str(round(v, 2)), ha='center', va='bottom', fontsize=10)
```

```
plt.title('Top 10 Customer Based on Sales')
plt.xlabel('Customer')
plt.ylabel('Sales')
plt.xticks(rotation=90)

plt.show()
```



Setelah kita analisa, kita mendapati:

1. Anthem sebagai Company dengan penjualan tertinggi.
2. Allianz sebagai peringkat kedua
3. Ford Motor sebagai peringkat ketiga.

Dengan data ini, kita dapat mengidentifikasi perusahaan-perusahaan utama dalam portofolio kita dan merencanakan program pemasaran yang berfokus pada mempertahankan pelanggan (Loyal Customer) di perusahaan-perusahaan tersebut.

---

Dalam tahap ini, kita mencari tahu penjualan berdasarkan segmen produk untuk perusahaan-perusahaan teratas dalam Top 10 Sales Company.

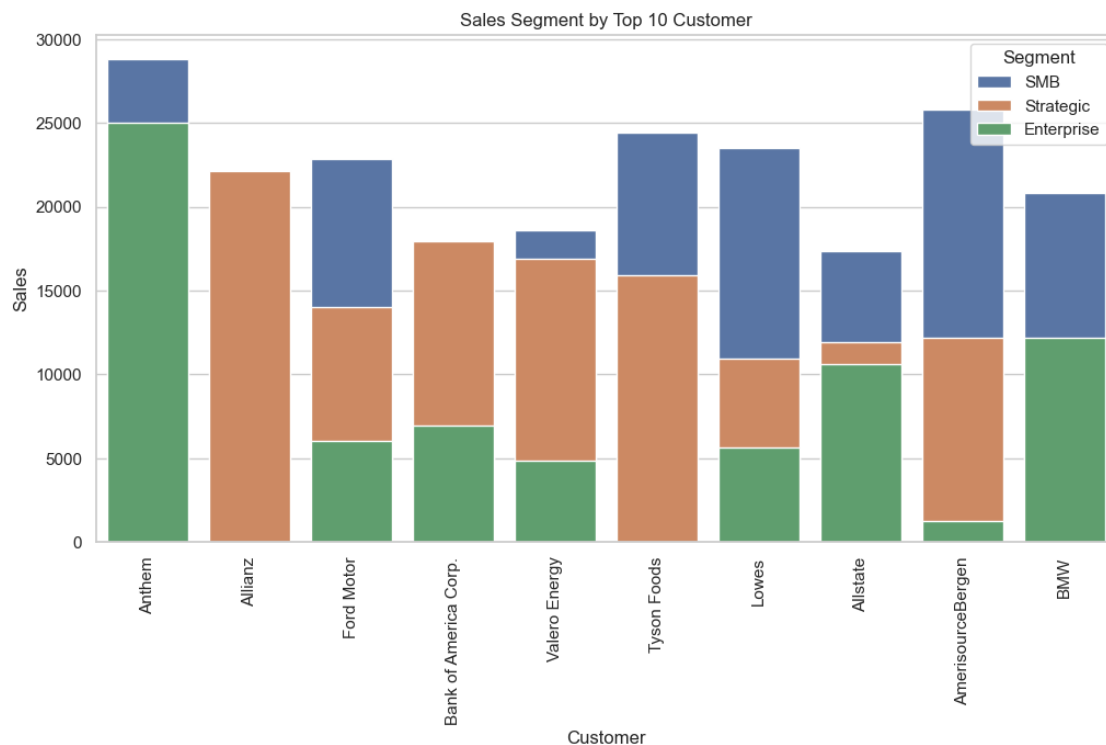
```
[34]: customer_sales = AWS.groupby('Customer')['Sales'].sum().reset_index().
      ↪sort_values(by='Sales', ascending=False).head(10)
```



```
plt.figure(figsize=(12,6))
sns.set(style="whitegrid", font_scale=1)
sns.barplot(data=AWS,x='Customer', y='Sales',estimator=np.sum,
            hue='Segment',ci=0,order=customer_sales['Customer'],dodge=False)

plt.title('Sales Segment by Top 10 Customer')
plt.xticks(rotation=90)

plt.show()
```



Hasilnya menunjukkan:

- Anthem didominasi oleh segmen SMB, diikuti oleh Enterprise.
- Allianz hanya membeli produk dari segmen Strategic.
- Ford Motor didominasi oleh SMB, diikuti oleh Strategic, dan terakhir Enterprise.

Dengan informasi ini, kita dapat berfokus pada pengembangan produk dan strategi pemasaran yang lebih sesuai dengan preferensi segmen dari masing-masing perusahaan tersebut. Selain itu, kita juga dapat mengambil langkah-langkah untuk mengumpulkan umpan balik dari mereka tentang kebutuhan mereka terhadap produk kami, sehingga kita dapat melakukan pengembangan produk yang lebih tepat sasaran dan meningkatkan kepuasan pelanggan.

### 3.7 Product

Dalam tahap ini, kita mencari data profit berdasarkan produk

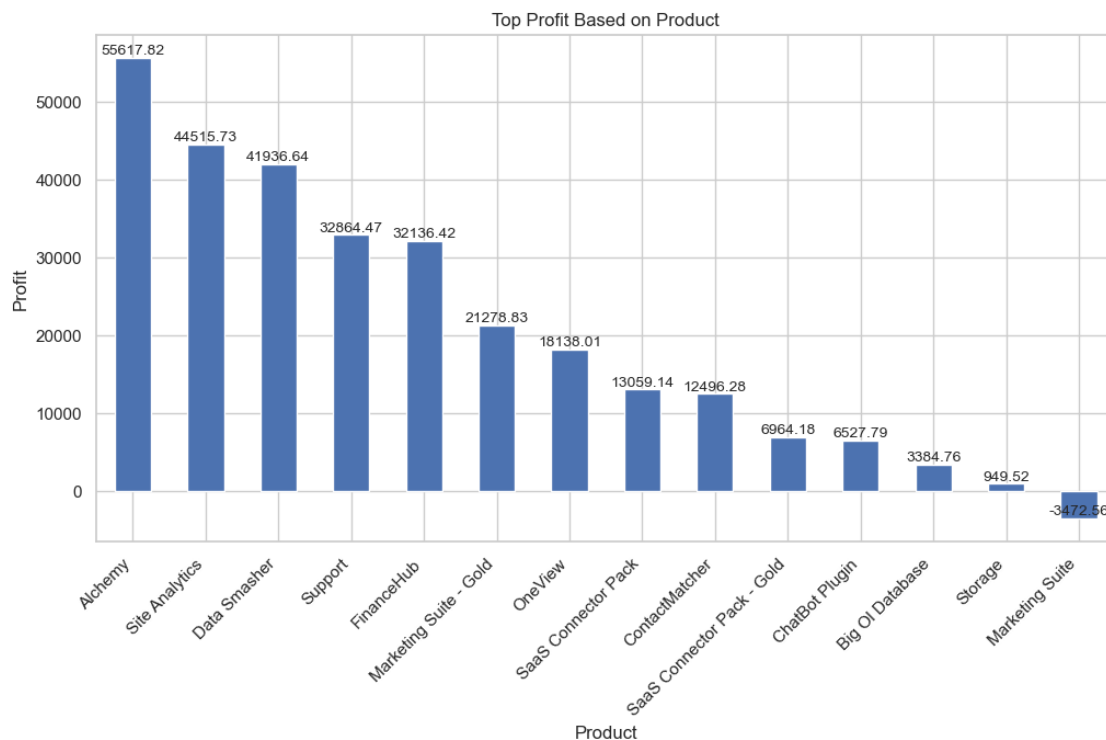
```
[35]: profit_quantity = AWS[['Product', 'Profit']].groupby('Product').sum().
      ↪sort_values(by='Profit', ascending=False)

# Membuat grafik batang
ax = profit_quantity.head(14).plot(kind='bar', legend=False, figsize=(12, 6))

# Menambahkan label pada batang
for i, v in enumerate(profit_quantity.head(14)['Profit']):
    ax.text(i, v + 50, str(round(v, 2)), ha='center', va='bottom', fontsize=10)

plt.title('Top Profit Based on Product')
plt.xlabel('Product')
plt.ylabel('Profit')
plt.xticks(rotation=45, ha='right')

plt.show()
```



Analisa ini memberikan wawasan berikut:

- Alchemy menduduki peringkat pertama dalam hal profit.
- Analytics berada di urutan kedua dalam hal profit.

- Data Smasher berada di urutan ketiga dalam hal profit.
- Marketing Suite adalah satu-satunya produk yang mengalami kerugian profit.

Dengan data ini, kita dapat melakukan fokus pemasaran dan pengembangan produk pada produk-produk yang menguntungkan.

Sementara itu, produk yang tidak menguntungkan dapat dievaluasi kembali, dan jika perlu, dipertimbangkan untuk dihapus dari portofolio produk kita. Ini akan membantu dalam mengoptimalkan kinerja bisnis dan alokasi sumber daya yang lebih efisien.

---

Langkah selanjut nya kita akan analisa product secara quantity yang terjual

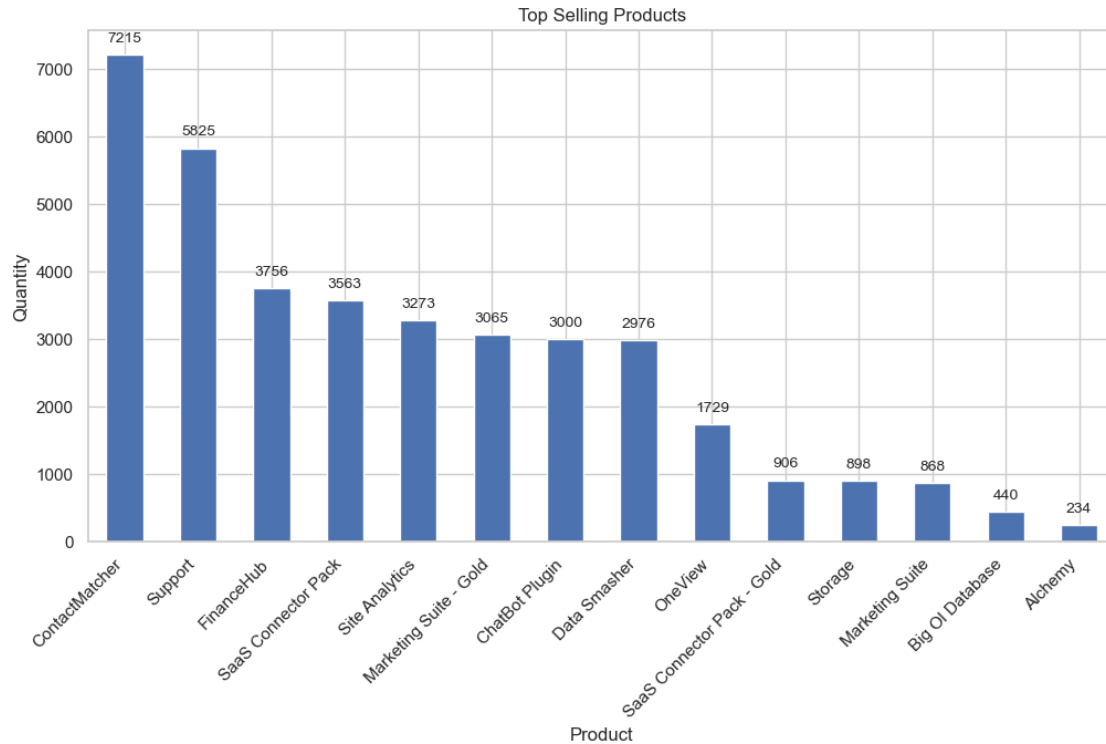
```
[36]: product_quantity = AWS[['Product', 'Quantity']].groupby('Product').sum().
      ↪sort_values(by='Quantity', ascending=False)

# Membuat grafik batang
ax = product_quantity.head(14).plot(kind='bar', legend=False, figsize=(12, 6))

# Menambahkan label pada batang
for i, v in enumerate(product_quantity.head(14)['Quantity']):
    ax.text(i, v + max(product_quantity['Quantity']) * 0.02, str(int(v)),
    ↪ha='center', va='bottom', fontsize=10)

plt.title('Top Selling Products')
plt.xlabel('Product')
plt.ylabel('Quantity')
plt.xticks(rotation=45, ha='right')

plt.show()
```



Selanjut nya kita akan analisa Rata-Rata Discount pada Product

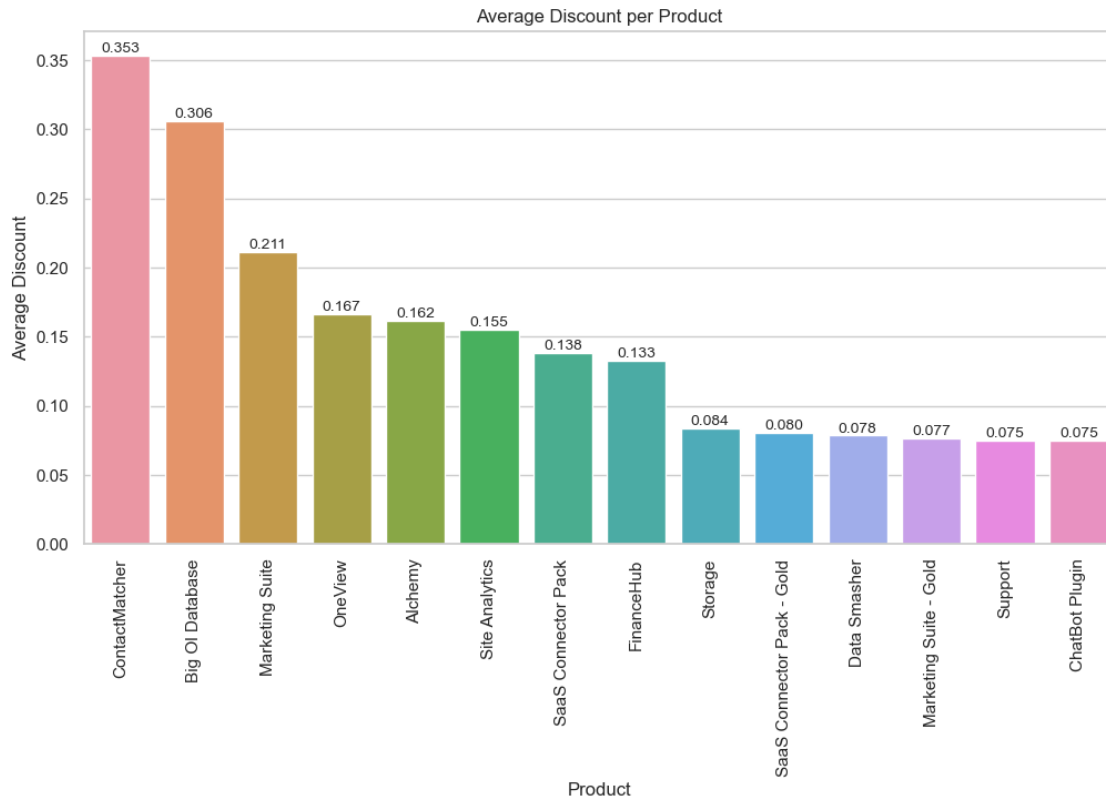
```
[37]: average_discount = AWS.groupby('Product')['Discount'].mean().reset_index().
      ↪sort_values(by='Discount', ascending=False)

# Membuat grafik batang
plt.figure(figsize=(12, 6))
ax = sns.barplot(data=average_discount, x='Product', y='Discount')

# Menambahkan label pada batang
for p in ax.patches:
    ax.annotate(f'{p.get_height():.3f}', (p.get_x() + p.get_width() / 2., p.
    ↪get_height()), ha='center', va='bottom', fontsize=10)

plt.title('Average Discount per Product')
plt.xlabel('Product')
plt.ylabel('Average Discount')
plt.xticks(rotation=90)

plt.show()
```



Berdasarkan analisis data, kita dapat menyimpulkan bahwa profit dari sebuah produk tidak hanya dipengaruhi oleh margin produk itu sendiri, tetapi juga dipengaruhi oleh rata-rata discount yang diberikan pada produk tersebut. Hal ini dapat dilihat dari fakta bahwa peringkat profit produk “ContactMatcher” mengalami penurunan, sementara produk “Alchemy” mengalami peningkatan.

Penurunan peringkat profit “ContactMatcher” mungkin terkait dengan adanya peningkatan rata-rata discount yang diberikan pada produk ini. Diskon yang lebih tinggi dapat mengurangi margin keuntungan produk, sehingga mempengaruhi profit keseluruhan. Sementara itu, peningkatan peringkat profit “Alchemy” mungkin terjadi karena penurunan rata-rata discount yang diberikan pada produk ini, yang dapat meningkatkan margin keuntungan dan, akhirnya, profit.

Dengan demikian, analisis ini menunjukkan bahwa pengelolaan discount dengan bijak dapat berdampak signifikan pada profitabilitas produk-produk tertentu dalam bisnis.

Dalam tahap ini, kita mencari data profit berdasarkan produk yang sudah dibagi berdasarkan segmennya.

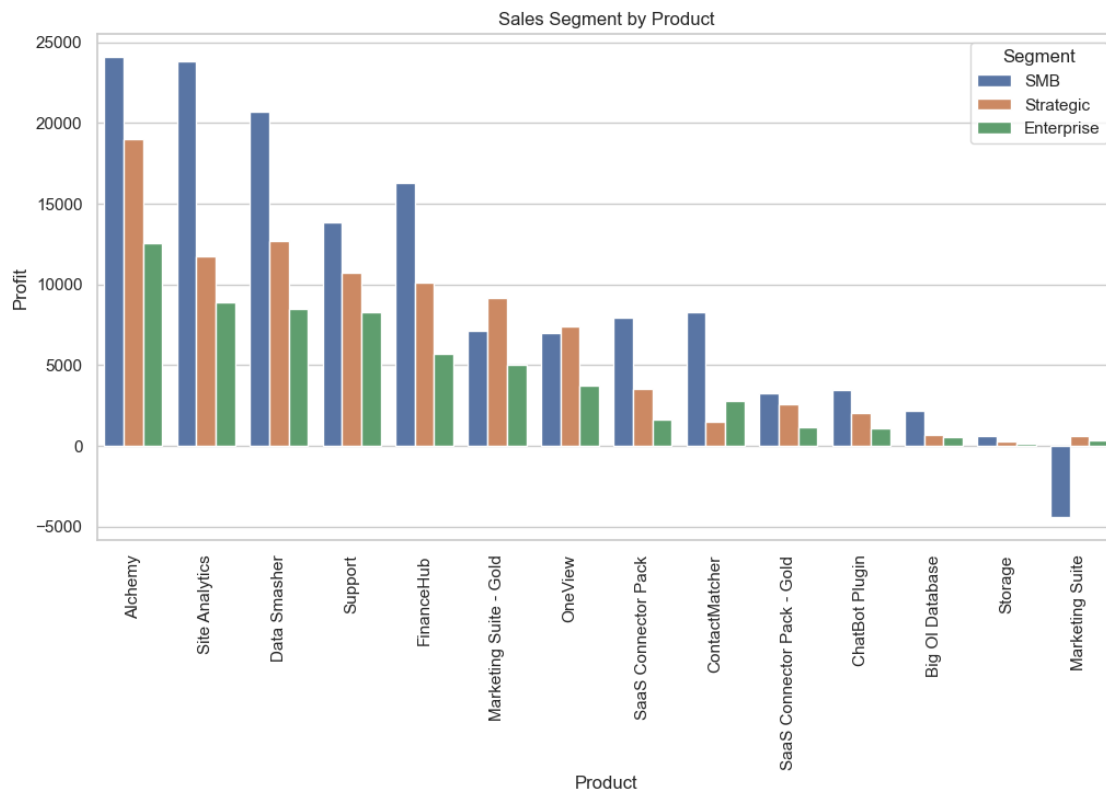
```
[38]: product_profit = AWS.groupby('Product')['Profit'].sum().reset_index().
      ↪sort_values(by='Profit', ascending=False)

plt.figure(figsize=(12,6))
sns.set(style="whitegrid", font_scale=1)
```

```
sns.barplot(data=AWS,x='Product', y='Profit',estimator=np.sum,
            hue='Segment',ci=0,order=product_profit['Product'])

plt.title('Sales Segment by Product')
plt.xticks(rotation=90)

plt.show()
```



Data ini memungkinkan kita untuk memiliki pengukuran yang lebih terperinci dan tepat sasaran dalam melakukan pengembangan produk dan pemasaran.

Selain itu, perihal **Marketing Suite**, kita dapat melakukan ulasan kembali atau pertimbangan untuk menghentikan produk tersebut, terutama dalam segmen SMB, jika hasilnya tidak memadai atau tidak menguntungkan. Langkah ini akan membantu dalam mengoptimalkan strategi dan portofolio produk kita sesuai dengan kebutuhan dan preferensi pelanggan dalam masing-masing segmen.

## 4 Conclusion And Recommendation

### 4.1 Conclusion

Berdasarkan analisis yang telah dilakukan, dapat disimpulkan bahwa ada beberapa aspek yang perlu diperhatikan untuk meningkatkan kinerja perusahaan.

- Penurunan signifikan dalam time to time sales dan profit sebagian besar disebabkan oleh kebijakan diskon dan strategi pemasaran yang kurang tepat sasaran. Adanya penurunan profit pada bulan salah satunya dikarenakan penggunaan diskon yang berlebihan, seperti yang kita ketahui meskipun discount dapat meningkatkan jumlah transaksi, discount juga dapat mengurangi profitabilitas perusahaan. Solusi potensial adalah mengalokasikan diskon dengan bijak, fokus pada bulan-bulan dengan transaksi yang dominan lebih sedikit untuk meningkatkan profit dan penjualan.
- Dengan memperhatikan data penjualan berdasarkan negara, kita dapat merancang strategi pemasaran yang lebih tepat sasaran. Negara dengan penjualan rendah dapat menjadi target untuk mencari pasar baru, sementara negara dengan penjualan tinggi perlu mendapatkan promosi khusus untuk mempertahankan dan meningkatkan kontribusi mereka terhadap perusahaan.
- Analisis segmentasi memberikan wawasan berharga. Pertama, dengan memeriksa penjualan berdasarkan industri, kita dapat menilai kinerja masing-masing segmen sesuai dengan industri yang ada. Dengan memahami mana segmen yang lebih kuat dalam industri tertentu, kita dapat menentukan segmen mana yang harus dikembangkan lebih lanjut untuk memanfaatkan peluang yang ada.
- Melalui analisis pelanggan atau perusahaan, kita dapat mengidentifikasi perusahaan-perusahaan utama yang menjadi pelanggan utama dan produk-produk yang paling banyak dibeli oleh mereka. Dengan informasi ini, kita dapat menentukan segmen mana yang perlu dikembangkan lebih lanjut untuk memenuhi kebutuhan setiap perusahaan tersebut. Ini akan membantu perusahaan dalam merancang strategi khusus untuk memperluas basis pelanggan dan meningkatkan penjualan di segmen yang relevan
- Analisis profit berdasarkan produk membantu kita mengidentifikasi produk-produk yang berkinerja baik dan yang kurang baik, serta menentukan segmen mana di mana produk tersebut tampil baik atau buruk. Dengan pengetahuan ini, kita dapat fokus pada mengoptimalkan produk-produk yang berkinerja baik dan mengembangkan strategi perbaikan untuk produk yang tidak berkinerja sebagaimana diharapkan. Selain margin pada produk, discount juga dapat mengurangi profitabilitas dari produk tersebut. Oleh karena itu, pemahaman yang mendalam tentang bagaimana discount memengaruhi profit menjadi kunci dalam pengambilan keputusan bisnis.

Secara keseluruhan, dengan memahami dan mengambil langkah-langkah berdasarkan analisis ini, perusahaan dapat meningkatkan profitabilitas, penjualan, dan strategi pemasaran mereka sesuai dengan potensi yang ada dalam berbagai segmen dan pasar.

## 4.2 Recommendation

- 1. Optimasi Kebijakan Diskon dan Strategi Pemasaran:** - **Peninjauan Kebijakan Diskon:** Evaluasi kebijakan diskon saat ini dan pastikan diskon dialokasikan secara bijak jangan menggunakan diskon berlebih pada bulan dengan penjualan yang tinggi seperti pada bulan: - **September - November - Desember - Penggunaan Diskon yang Tepat:** Pertimbangkan penggunaan diskon pada bulan-bulan dengan penjualan rendah, untuk meningkatkan profitabilitas seperti pada bulan: - **Januari - Februari - Maret**
- 2. Penyesuaian Strategi Berdasarkan Data Penjualan Negara:** - **Penjualan di Negara dengan Potensi Rendah:** Fokus pada negara dengan penjualan rendah seperti **Qatar, Iceland,**

**Denmark, Croatia, dan Slovenia** untuk mencari peluang pasar baru. - **Penjualan di Negara dengan Potensi Tinggi:** Tingkatkan promosi di negara dengan penjualan tinggi seperti **AS, Inggris, Jepang, Kanada, dan Prancis** untuk mempertahankan kontribusi mereka.

**3. Pengembangan Segmen yang Potensial:** - **Segmentasi Berdasarkan Industri:** Identifikasi segmen yang memiliki potensi pertumbuhan, seperti : - Segment **SMB** pada industri **Finance, Energy, Manufacturing, Healthcare, Tech, Consumer Product, Retail, dan Transportation** - Segment **Strategic** pada industri **Communication** - Segment **Enterprise** pada industri **Misc**

**4. Fokus pada Perusahaan Utama dan Meminta Feedback:** - **Strengthen Hubungan dengan Perusahaan Utama:** Perkuat hubungan dengan perusahaan utama seperti: - **Anthem** - **Allianz** - **Ford Motor** - **Pengembangan Produk Khusus:** Kembangkan produk yang sesuai dengan kebutuhan perusahaan utama, seperti: - Segment **SMB** pada **Anthem** dan **Ford** - Segment **Strategic** pada **Allianz** - **Umpan Balik Aktif:** Selalu minta umpan balik dari perusahaan-perusahaan utama untuk memahami kebutuhan mereka dan menyesuaikan produk dan layanan.

**5. Optimasi Portofolio Produk:** - **Analisis Profit Berdasarkan Produk:** Evaluasi profitabilitas produk berdasarkan segment mereka, seperti: - Segment **SMB** pada **Alchemcy** dan **Site Analytics** - Segment **Strategic** pada **Data Smasher**. - **Penggunaan Diskon yang Bijak:** Hindari penggunaan diskon yang berlebihan, terutama pada produk dengan quantity penjualan yang tinggi seperti pada product: - **Contact Matcher** - **Big Ol Database** - **Evaluasi Produk yang Tidak Efektif:** Evaluasi kembali dan pertimbangkan untuk menghapus produk yang tidak dapat mencapai kinerja yang diharapkan dari portofolio untuk menghindari pemborosan sumber daya seperti: - **Marketing Suites** - **Storage**

**6. Monitoring dan Evaluasi Terus-Menerus:** - **Pemantauan Rutin:** Implementasikan pemantauan dan evaluasi terus-menerus untuk memastikan perbaikan berkelanjutan dalam kinerja perusahaan.