EDA dengan Visualisasi Data

DQLab LiveClass

Outline

- Konsep data visualisasi data dan storytelling
- Summarize data dengan pivot table
- Visualisasi data dengan seaborn

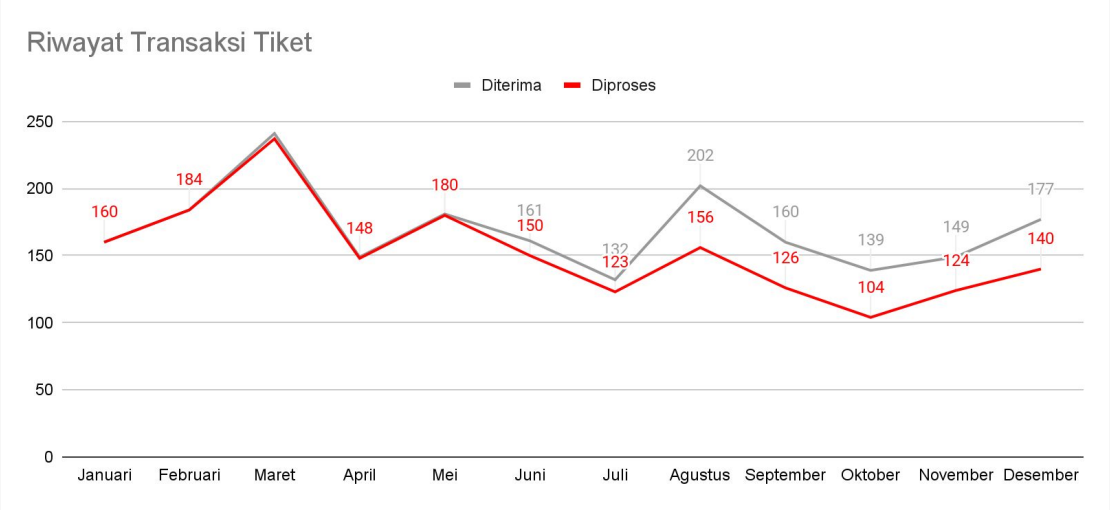
Perhatikan data berikut

Bulan	Diterima	Diproses
Januari	160	160
Februari	184	184
Maret	241	237
April	149	148
Mei	181	180
Juni	161	150
Juli	132	123
Agustus	202	156
September	160	126
Oktober	139	104
November	149	124
Desember	177	140

Sebuah perusahaan penjualan tiket merilis rekapan transaksi yang berisi jumlah permintaan tiket yang diterima dan yang berhasil diproses setiap bulannya.

Insight apa yang kamu dapatkan?

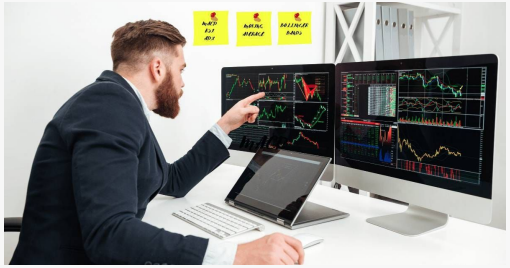
Sekarang perhatikan chart di berikut



Insight apa yang kamu dapatkan?
Apa bedanya dengan melihat angka-angka pada tabel?

Konsep visualisasi data dan Storytelling

Menyederhanakan data yang membingungkan



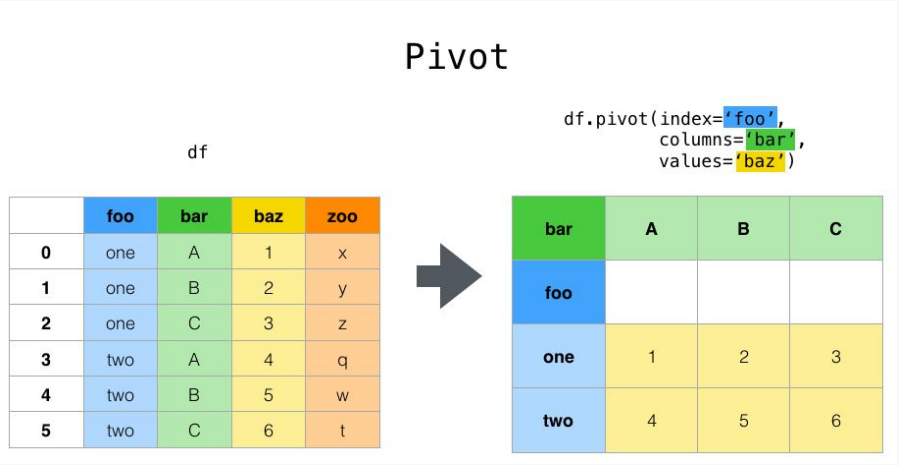
Mengenali kejadian berulang (pattern) untuk digunakan dalam *forecasting*

Mendapatkan informasi penting (insights)

Simplifying complex information into engaging story and presenting it visually enables decision-makers to make informed and effective decisions quickly and accurately.

Pivot Table

- Sebelum membuat visualisasi, kita harus membuat beberapa summary dari data
- Summary data digunakan untuk mencari informasi sebanyak mungkin serta menguji beberapa hipotesis
- Summary data membantu kita memilah informasi mana yang penting dan menjawab permasalahan
- Metode yang umum dalam membuat summary data adalah pivot table



	foo	bar	baz	zoo
0	one	A	1	x
1	one	B	2	y
2	one	C	3	z
3	two	A	4	q
4	two	B	5	w
5	two	C	6	t

bar	A	B	C
foo			
one	1	2	3
two	4	5	6

Komponen pivot table

Komponen	Deskripsi
Data	Data yang akan dibuat summarynya
Index/Row & Columns	Baris dan kolom untuk menentukan bagaimana data ditampilkan
Values	Nilai yang akan dihitung
Aggregate Functions	Fungsi hitung

Syntax:

```
pd.pivot_table(
        data=df,
        index='kolom_a',
        columns='kolom_b',
        values='kolom_c',
        aggfunc=<nama_fungsi>
)
```

Fungsi agregat yang sering digunakan:
sum, mean, min, max, count

Contoh pivot table (1)

```
pd.pivot_table(
    data=df,
    index='Category',
    values='Sales',
    aggfunc='sum'
)
```

Category	Sales
Furniture	741999.7953
Office Supplies	719047.0320
Technology	836154.0330

Jumlah pendapatan (sales) berdasarkan kategori produk yang dijual

Contoh pivot table (2)

```
pd.pivot_table(
    data=df,
    index='Category',
    columns='Region',
    values='Sales',
    aggfunc='sum'
)
```

Region	Central	East	South	West
Category				
Furniture	163797.1638	208291.204	117298.684	252612.7435
Office Supplies	167026.4150	205516.055	125651.313	220853.2490
Technology	170416.3120	264973.981	148771.908	251991.8320

Jumlah pendapatan berdasarkan kategori produk dan wilayah

Contoh pivot table (3)

```
pd.pivot_table(
    data=df,
    index='Category',
    columns=['Region','Segment'],
    values='Sales',
    aggfunc='sum'
)
```

Region	Central			East			South			West		
Segment	Consumer	Corporate	Home Office	Consumer	Corporate	Home Office	Consumer	Corporate	Home Office	Consumer	Corporate	Home Office
Category												
Furniture	86229.219	52085.6018	25482.343	114211.802	64209.046	29870.356	70800.204	29645.0315	16853.4485	119808.087	83080.1065	49724.550
Office Supplies	93111.479	41137.7010	32777.235	101255.136	66474.735	37786.184	59504.581	45930.1700	20216.5620	110080.940	77133.8560	33638.453
Technology	72690.736	64772.5100	32953.066	135441.229	69725.566	59807.186	65276.186	46310.7310	37184.9910	132991.746	65641.3120	53358.774

Index atau kolom dapat dibuat bertingkat dengan memasukkan nama variable ke dalam list

Peserta dipersilahkan mencoba pivot_table

Visualisasi data dengan Seaborn

- Seaborn adalah salah satu library python yang berfokus pada visualisasi data
- Seaborn dibangun di atas matplotlib sekaligus menyederhanakan syntax matplotlib
- `import seaborn as sns` untuk menggunakan seaborn

Line chart

- Line chart biasanya digunakan untuk melihat trend atau perubahan dari waktu ke waktu
- Sumbu x pada line chart biasanya adalah kolom dengan tipe data yang memiliki urutan, contohnya: tanggal
- Syntax:

```
sns.lineplot(data, sumbu_x, sumbu_y)
```

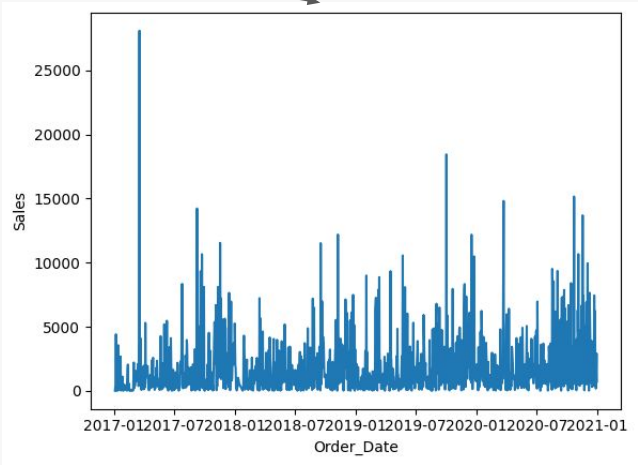
Contoh: Line Chart Single

```
[8]  data = pd.pivot_table(
         data=df,
         index='Order_Date',
         values='Sales',
         aggfunc='sum'
     ).reset_index()

     data.head()
```

	Order_Date	Sales
0	2017-01-03	16.448
1	2017-01-04	288.060
2	2017-01-05	19.536
3	2017-01-06	4407.100
4	2017-01-07	87.158

```
[9]  sns.lineplot(
         data=data,
         x='Order_Date',
         y='Sales'
         )
```



Multiple line chart

- Untuk membuat multiple line chart, siapkan satu kolom yang menunjukkan pembagian line chart
- Masukkan kolom tersebut ke dalam parameter hue pada fungsi sns.linechart
- Syntax

```
sns.lineplot(data, x, y, hue)
```

Contoh: Multiple Line Chart

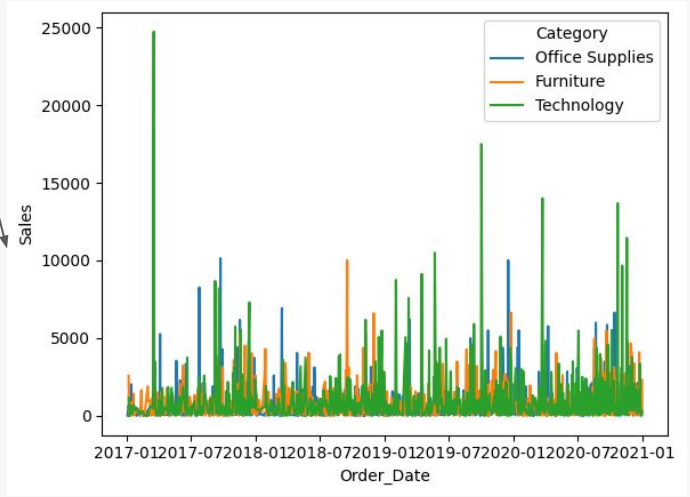
```
# Multiple line

data = pd.pivot_table(
    data=df,
    index=['Order_Date','Category'],
    values='Sales',
    aggfunc='sum'
).reset_index()

data.head()
```

	Order_Date	Category	Sales
0	2017-01-03	Office Supplies	16.448
1	2017-01-04	Office Supplies	288.060
2	2017-01-05	Office Supplies	19.536
3	2017-01-06	Furniture	2573.820
4	2017-01-06	Office Supplies	685.340

```
[11] sns.lineplot(
        data=data,
        x='Order_Date',
        y='Sales',
        hue='Category'
        )
```



Bar chart

- Bar chart digunakan untuk membandingkan nilai antar variabel
- Sumbu x pada bar chart tidak perlu merupakan variabel dengan urutan
- Syntax:

```
sns.barplot(data, sumbu_x, sumbu_y)
```

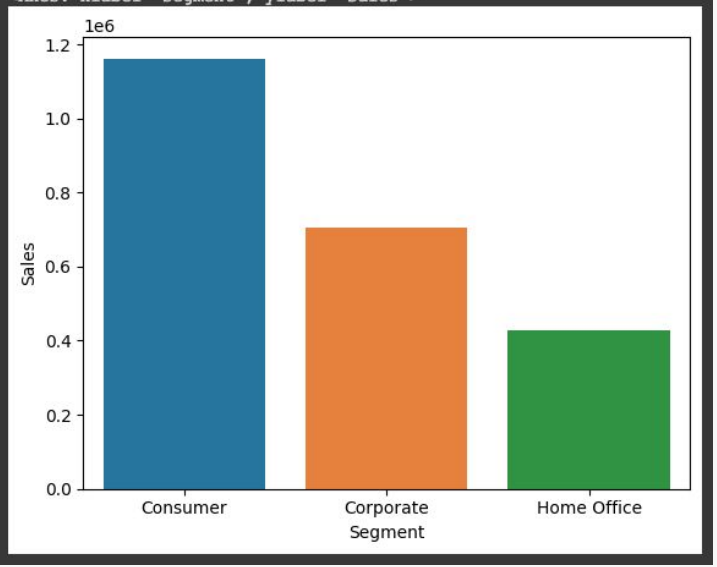
```
[12] data = pd.pivot_table(
        data=df,
        index='Segment',
        values='Sales',
        aggfunc='sum'
    ).reset_index()

    data.head()
```

	Segment	Sales
0	Consumer	1.161401e+06
1	Corporate	7.061464e+05
2	Home Office	4.296531e+05

```
[13] sns.barplot(data=data, x='Segment', y='Sales')
```

```
<Axes: xlabel='Segment', ylabel='Sales'>
```



Cluster Bar Chart

- Selain membuat barchart sederhana, kita juga dapat membuat breakdown dari barchart ke dalam komponennya
- Masukkan variabel yang akan menjadi komponennya ke dalam parameter hue
- Syntax:

```
sns.barplot(data, sumbu_x, sumbu_y, hue)
```

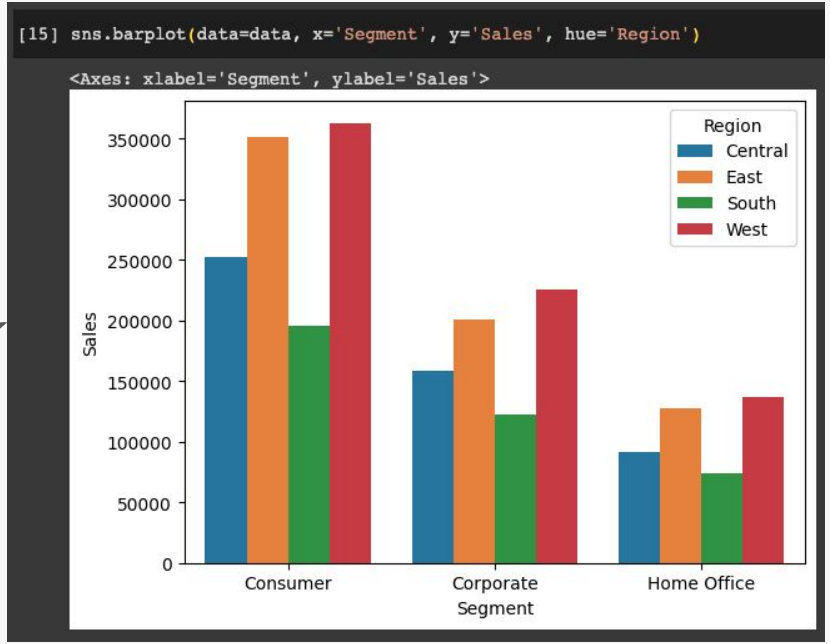
Cluster Bar Chart

```
data = pd.pivot_table(
    data=df,
    index=['Segment','Region'],
    values='Sales',
    aggfunc='sum'
).reset_index()

data.head()
```

	Segment	Region	Sales
0	Consumer	Central	252031.4340
1	Consumer	East	350908.1670
2	Consumer	South	195580.9710
3	Consumer	West	362880.7730
4	Corporate	Central	157995.8128

```
[15] sns.barplot(data=data, x='Segment', y='Sales', hue='Region')
```



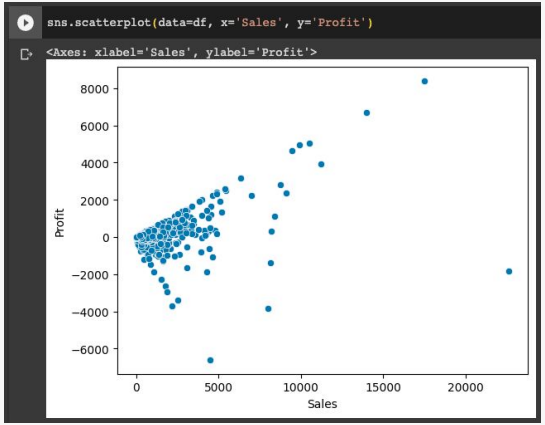
Scatterplot

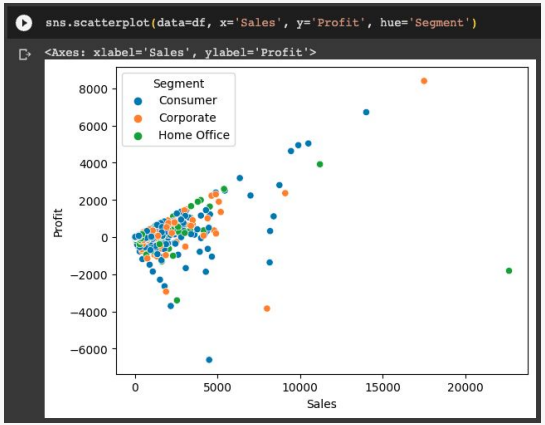
- Scatterplot digunakan untuk melihat korelasi atau hubungan antar dua variabel numerik
- Syntax

  sns.scatterplot(data, x, y)

- Atau

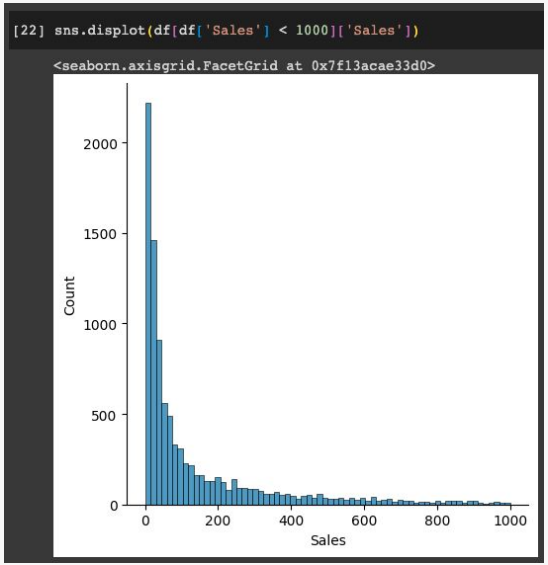
  sns.scatterplot(data, x, y, hue)





Displot

- Displot digunakan untuk menampilkan distribusi dari series numerik
- Secara default displot akan menampilkan histogram
- Syntax: sns.displot(<series>)



Menampilkan distribusi dari Sales dengan sales di bawah 1000

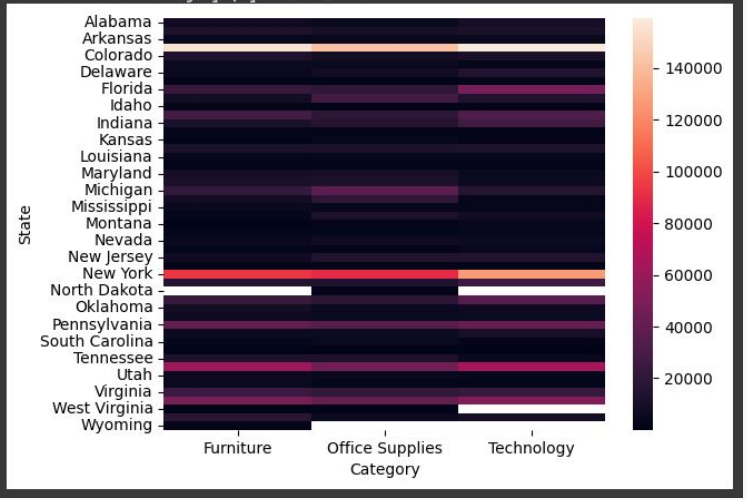
Heatmap

- Heatmap memudahkan pembacaan tabel dengan cara memberikan warna pada cell berdasarkan nilai pada cell tersebut
- Syntax: sns.heatmap(tabel)

```
[25] data = pd.pivot_table(
        data=df,
        index='State',
        columns='Category',
        values='Sales',
        aggfunc='sum'
    )
```

```
sns.heatmap(data)
```

```
<Axes: xlabel='Category', ylabel='State'>
```



DQLab

live Class

Terimakasih!

dqlab www.dqlab.id