

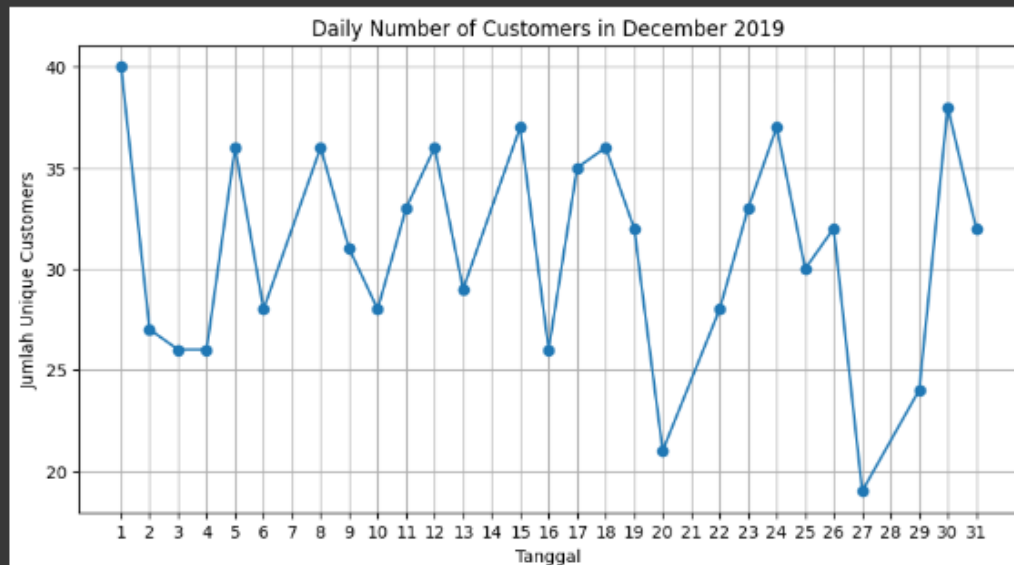
## Mini Projec

```
[2] 1 import pandas as pd
    2 import matplotlib.pyplot as plt
    3 import pandas as pd
    4 import matplotlib.pyplot as plt
    5
    6 # Load dataset
    7 df = pd.read_csv('retail_raw_reduced.csv')
    8
    9 # Mengubah kolom order_date menjadi tipe datetime
   10 df['order_date'] = pd.to_datetime(df['order_date'])
   11
   12 # Filter data untuk bulan Desember 2019
   13 dataset_december = df[(df['order_date'].dt.month == 12) & (df['order_date'].dt.year == 2019)]
```

**jumlah pembeli harian (daily number of customers) selama bulan Desember.**

Jumlah pembeli harian (daily number of customers) selama bulan Desember.

```
1 # Jumlah pembeli harian (daily number of customers) selama bulan Desember.
2
3 # Menghitung jumlah unique customers di tiap tanggal
4 daily_customers = dataset_december.groupby(dataset_december['order_date'].dt.day)['customer_id'].nunique()
5
6 # Membuat line chart jumlah unique customers
7 plt.figure(figsize=(10, 5))
8 plt.plot(daily_customers.index, daily_customers.values, marker='o')
9 plt.title('Daily Number of Customers in December 2019')
10 plt.xlabel('Tanggal')
11 plt.ylabel('Jumlah Unique Customers')
12 plt.xticks(range(1, 32))
13 plt.grid(True)
14 plt.show()
```



## Case 1

### Case 1:

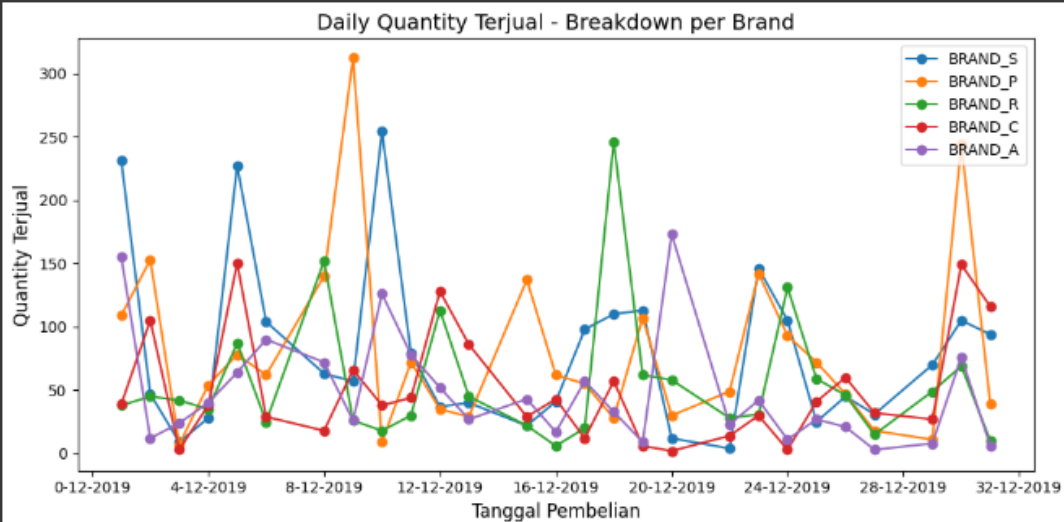
```
1 #Case 1
2 # Menghitung total kuantitas penjualan per brand di bulan Desember 2019
3 top5_brands = dataset_december.groupby('brand')['quantity'].sum().nlargest(5)
4
5 # Membuat variabel dataset_top5brand_dec yang berisi data penjualan bulan Desember 2019 hanya untuk top 5 brand
6 dataset_top5brand_dec = dataset_december[dataset_december['brand'].isin(top5_brands.index)]
7
8 # Menampilkan top 5 brands beserta quantity terjual
9 print("Top 5 brands berdasarkan quantity terjual di bulan Desember 2019:")
10 for i, (brand, quantity) in enumerate(zip(top5_brands.index, top5_brands.values), 1):
11     print(i, ".", brand, "(quantity:", quantity, ")")
```

```
Top 5 brands berdasarkan quantity terjual di bulan Desember 2019:
1 . BRAND_S (quantity: 2197 )
2 . BRAND_P (quantity: 2194 )
3 . BRAND_R (quantity: 1588 )
4 . BRAND_C (quantity: 1365 )
5 . BRAND_A (quantity: 1315 )
```

## Case 2

Case 2: Multi-line chart daily quantity untuk brand top 5

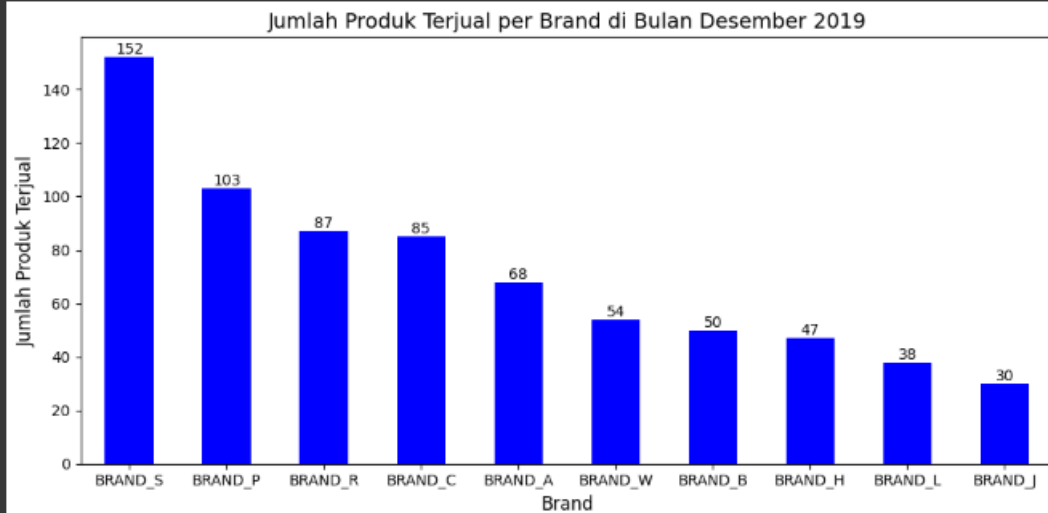
```
1 #Case 2
2
3 # Group data berdasarkan tanggal dan brand, lalu hitung jumlah quantity
4 daily_quantity = dataset_december.groupby(['order_date', 'brand'])['quantity'].sum().unstack()
5
6 # Filter data untuk top 5 brand dengan jumlah quantity tertinggi
7 top_5_brands = daily_quantity.sum().nlargest(5).index
8 daily_quantity_top5 = daily_quantity[top_5_brands]
9
10 # Menggambar multi-line chart
11 fig, ax = plt.subplots(figsize=(10, 5))
12
13 for column in daily_quantity_top5.columns:
14     ax.plot(daily_quantity_top5.index.day, daily_quantity_top5[column], marker='o', label=column)
15
16 # Mencari tanggal dengan lonjakan (jumlah quantity terjual paling tinggi)
17 date_of_spike = daily_quantity_top5.sum(axis=1).idxmax()
18 max_quantity = daily_quantity_top5.sum(axis=1).max()
19
20 # Menambahkan anotasi untuk titik lonjakan
21 ax.annotate('Lonjakan', xy=(date_of_spike.day, max_quantity), xytext=(date_of_spike.day, max_quantity + 5),
22            arrowprops=dict(facecolor='yellow', arrowstyle='->'))
23
24 # Menambahkan judul dan label sumbu
25 ax.set_title('Daily Quantity Terjual - Breakdown per Brand', fontsize=14)
26 ax.set_xlabel('Tanggal Pembelian', fontsize=12)
27 ax.set_ylabel('Quantity Terjual', fontsize=12)
28
29 # Menampilkan legenda
30 ax.legend()
31
32 # Mengatur format tanggal pada sumbu x
33 ax.xaxis.set_major_locator(plt.MaxNLocator(10))
34 ax.xaxis.set_major_formatter(plt.FuncFormatter(lambda x, pos: f'{int(x)}-12-2019'))
35
36 # Menampilkan chart
37 plt.tight_layout()
38 plt.show()
```



## Case 3

Case 3:

```
1 #Case 3
2
3 # Menghitung jumlah produk terjual untuk setiap merek (brand) dan mengurutkannya
4 brand_product_count = dataset_december.groupby('brand')['product_id'].nunique()
5 brand_product_count_sorted = brand_product_count.sort_values(ascending=False)
6
7 # Menampilkan jumlah produk terjual per brand di bulan Desember 2019 dalam bentuk bar chart
8 plt.figure(figsize=(10, 5))
9 brand_product_count_sorted.plot(kind='bar', color='blue')
10
11 # Mengatur judul dan label sumbu
12 plt.title('Jumlah Produk Terjual per Brand di Bulan Desember 2019', fontsize=14)
13 plt.xlabel('Brand', fontsize=12)
14 plt.ylabel('Jumlah Produk Terjual', fontsize=12)
15
16 # Mengatur miring (diagonal) untuk label brand
17 plt.xticks(rotation=0, ha='center')
18
19 # Menampilkan nilai di atas setiap bar
20 for i, value in enumerate(brand_product_count_sorted.values):
21     plt.text(i, value, str(value), ha='center', va='bottom')
22
23 # Menampilkan bar chart
24 plt.tight_layout()
25 plt.show()
```

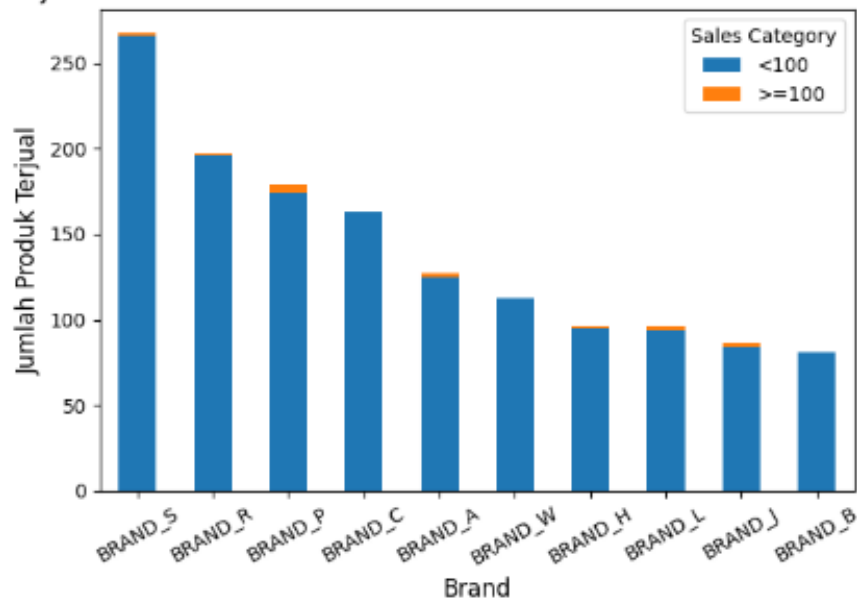


## Case 4

Case 4: Penjualan produk diatas 100 dan dibawah 100 selama Desember 2019

```
1 #Case 4
2
3 # Menghitung jumlah produk terjual per brand
4 brand_product_count = dataset_december.groupby('brand')['product_id'].count()
5
6 # Membuat kolom baru untuk kategori penjualan (>=100 atau <100)
7 dataset_december['sales_category'] = dataset_december['quantity'].apply(lambda x: '>=100' if x >= 100 else '<100')
8
9 # Menghitung jumlah produk terjual per brand dan kategori penjualan
10 brand_sales_category_count = dataset_december.groupby(['brand', 'sales_category']).size().unstack()
11
12 # Mengurutkan brand berdasarkan jumlah produk terjual secara keseluruhan
13 brand_sales_category_count = brand_sales_category_count.reindex(rank_brand_product_count.sort_values(ascending=False).index)
14
15 # Membuat stacked chart
16 plt.figure(figsize=(10, 5))
17 brand_sales_category_count.plot(kind='bar', stacked=True)
18
19 # Mengatur miring (diagonal) untuk label brand
20 plt.xticks(rotation=30, ha='center')
21
22 # Mengatur judul dan label sumbu
23 plt.title('Penjualan Produk diatas 100 dan dibawah 100 di Bulan Desember 2019', fontsize=14)
24 plt.xlabel('Brand', fontsize=12)
25 plt.ylabel('Jumlah Produk Terjual', fontsize=12)
26
27 # Menampilkan legenda
28 plt.legend(title='Sales Category')
29
30 # Menampilkan stacked chart
31 plt.tight_layout()
32 plt.show()
```

Penjualan Produk diatas 100 dan dibawah 100 di Bulan Desember 2019



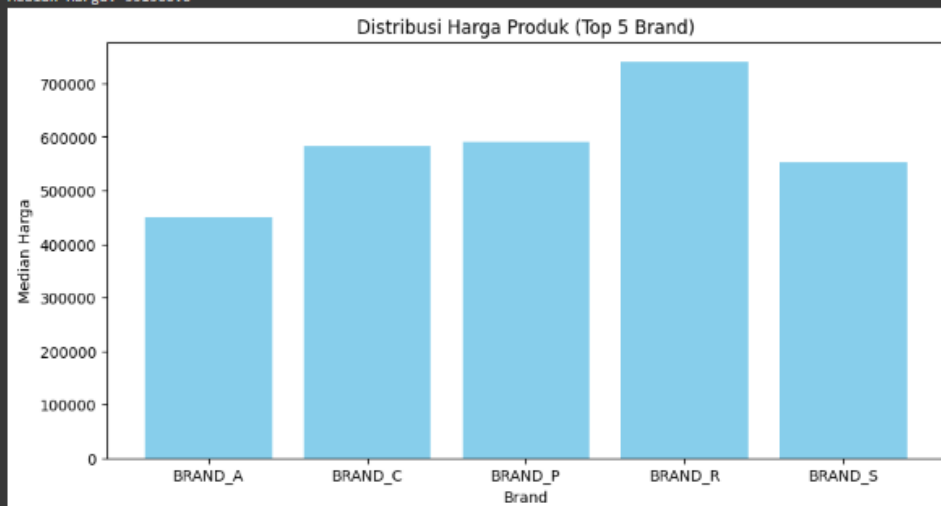
## Case 5

Case 5: Murah atau mahalkah harga produk brand top 5

```
1 # Case 5
2 # Menghitung median harga produk untuk setiap product_id dalam dataset_top5brand_dec
3 median_harga_produk = dataset_top5brand_dec.groupby('brand')['item_price'].median()
4
5 plt.figure(figsize=(10, 5))
6 plt.bar(median_harga_produk.index, median_harga_produk.values, color='skyblue')
7
8 # Mengatur label dan judul
9 plt.xlabel('Brand')
10 plt.ylabel('Median Harga')
11 plt.title('Distribusi Harga Produk (Top 5 Brand)')
12
13 # Menampilkan data brand dan median harganya
14 print("Data Brand dan Median Harga:")
15 for brand, median_harga in median_harga_produk.items():
16     print("Brand:", brand)
17     print("Median Harga:", median_harga)
18 # Menampilkan histogram
19 plt.show()
20
21 brand_termurah = median_harga_produk.idxmin()
22 brand_termahal = median_harga_produk.idxmax()
23 print()
24 print("Jika dilihat dari median harga, dapat dikatakan bahwa ", brand_termurah, "memiliki harga produk yang lebih murah dibandingkan dengan brand lainnya,")
25 print("sedangkan", brand_termahal, "memiliki harga produk yang lebih mahal dibandingkan dengan brand lainnya.")
```

Data Brand dan Median Harga:

Brand: BRAND\_A  
Median Harga: 450000.0  
Brand: BRAND\_C  
Median Harga: 583000.0  
Brand: BRAND\_P  
Median Harga: 590000.0  
Brand: BRAND\_R  
Median Harga: 740000.0  
Brand: BRAND\_S  
Median Harga: 551500.0

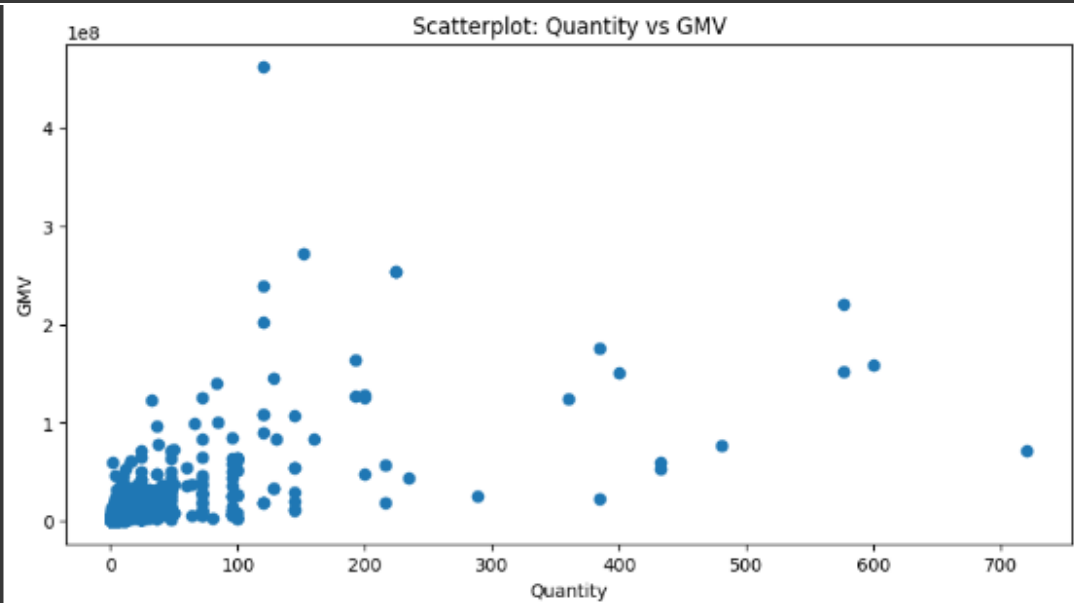


jika dilihat dari median harga, dapat dikatakan bahwa BRAND\_A memiliki harga produk yang lebih murah dibandingkan dengan brand lainnya, sedangkan BRAND\_R memiliki harga produk yang lebih mahal dibandingkan dengan brand lainnya.

## Case 6a

Case 6a: Korelasi quantity vs GMV

```
1 #Case 6a
2
3 # Menghitung GMV (quantity * item_price)
4 df['GMV'] = df['quantity'] * df['item_price']
5
6 # quantity vs GMV
7 plt.figure(figsize=(10, 5))
8 plt.scatter(df['quantity'], df['GMV'])
9 plt.xlabel('Quantity')
10 plt.ylabel('GMV')
11 plt.title('Scatterplot: Quantity vs GMV')
12 plt.show()
13
14 correlation_quantity_gmv = df['quantity'].corr(df['GMV'])
15 print("Korelasi antara quantity dan GMV:", correlation_quantity_gmv)
```



Korelasi antara quantity dan GMV: 0.6269420804799786

## Case 6b

Case 6b: Korelasi median harga vs quantity

```
1 #Case 6b
2 #Median harga vs quantity
3 median_harga = df.groupby('product_id')['item_price'].median()
4
5 df['median_harga'] = df['product_id'].map(median_harga)
6
7 plt.figure(figsize=(10, 6))
8 plt.scatter(df['median_harga'], df['quantity'])
9 plt.xlabel('Median Harga')
10 plt.ylabel('Quantity')
11 plt.title('Scatterplot: Median Harga vs Quantity')
12 plt.show()
13
14 correlation_medianharga_quantity = df['median_harga'].corr(df['quantity'])
15 print("Korelasi antara median harga dan quantity:", correlation_medianharga_quantity)
16
17
```

