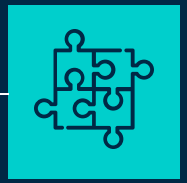
Analysis Brazilian E-Commerce Public Data

Oleh : M.Irfansyah

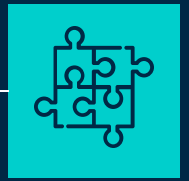
TABLE OF CONTENTS



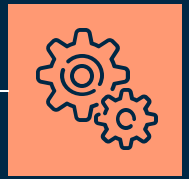
01 Rumusan Masalah

02 Data Wrangling

03 Data Cleaning



04 EDA



05 Visualitation & Explanatory Analysis

06 Kesimpulan

Rumusan Masalah

01

Rumusan Masalah

Terdapat tiga rumusan masalah:

1. Apa kategori barang yang paling banyak dibeli dan paling sedikit diminati?
2. Bulan apa yang terjadi penjualan tertinggi?
3. Hari apa yang sering digunakan oleh pembeli untuk melakukan transaksi?

Data Wranglin g

02

```
# import packages packages yang akan digunakan
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Melakukan import packages yang digunakan

```
customers = pd.read_csv("customers_dataset.csv")
geolocation = pd.read_csv("geolocation_dataset.csv")
order_items = pd.read_csv("order_items_dataset.csv")
order_payments = pd.read_csv("order_payments_dataset.csv")
orders = pd.read_csv("orders_dataset.csv")
product_name = pd.read_csv("product_category_name_translation.csv")
products = pd.read_csv("products_dataset.csv")
sellers = pd.read_csv("sellers_dataset.csv")
```

Melakukan import data yang akan digunakan untuk proses analisis data

```
product_name_df = pd.merge(
    left=products,
    right=product_name,
    how="left",
    left_on="product_category_name",
    right_on="product_category_name"
)
product_name_df.head()
```

Menggabungkan data products dengan product_name dan hasil gabungan kedua data diberi nama product_name

```
df_product_name = product_name_df[["product_id","product_category_name","product_category_name_english"]]
df_product_name.head()
```

Mendefinisikan sebuah data frame baru yang tersusun dari kolom product_id, product_category_name dan product_category_name_english yang diberi nama df_product_name

```
[ ]  order_product = pd.merge(
        left= order_items,
        right=df_product_name,
        how="left",
        left_on="product_id",
        right_on="product_id"
    )
    order_product.head()
```

Menggabungkan df_product_name dengan order_items.Hasil dari gabungan tersebut, kemudian digabungkan dengan data sellers.Hasil gabungan ini didefinisikan dengan seller_product

```
    seller_product = pd.merge(
        left= order_product,
        right=sellers,
        how="left",
        left_on="seller_id",
        right_on="seller_id"
    )
    seller_product.head()
```

```
[ ]  geolocation = geolocation.drop_duplicates
     (subset=['geolocation_zip_code_prefix'])
     geolocation.head()
```

Melakukan drop duplikat pada kolom geolocation_zip_code_prefix

```
[ ] df_order = seller_product.merge(geolocation, left_on='seller_zip_code_prefix',
                                    right_on='geolocation_zip_code_prefix', how='left')
    df_order.head()
```

Menggabungkan geolocation dengan seller_product .Hasil dari proses ini definisikan dengan df_order

```
[ ] df_order = df_order.drop(columns = ['geolocation_city','geolocation_state'])
```

Melakukan drop kolom geolocation_city dan geolocation_state pada data frame df_order

```
orders_payments = pd.merge(
    left= orders,
    right= order_payments,
    how="left",
    left_on="order_id",
    right_on="order_id"
)
orders_payments.head()
```

Menggabungkan orders dengan order_payments .Hasil dari proses ini definisikan dengan orders_payments

```
[ ]  orders_payments = orders_payments.drop
     (columns = ['payment_sequential','payment_installments'])
```

Melakukan drop kolom payment_sequential dan payment_installments pada data orders_payments

```
[ ]  customers = customers.drop(columns = ['customer_unique_id'])
```

Melakukan drop kolom customer_unique_id pada data customers

```
order_customers = orders_payments.merge(customers,
left_on='customer_id', right_on='customer_id',how='left')
order_customers.head()
```

Menggabungkan orders_payments dengan customers.Hasil dari proses ini definisikan dengan orders_customers

```
order =  order_customers.merge(geolocation, left_on='customer_zip_code_prefix',
right_on='geolocation_zip_code_prefix',how='left')
order.head()
```

Menggabungkan order_customers dengan geolocation.Hasil dari proses ini definisikan dengan order

```
order =  order.drop(columns = ['geolocation_city','geolocation_state'])
```

Melakukan drop kolom geolocation_city dan geolocation_ state pada data order

Didapat dua data frame baru yaitu df_order dan order

Data Cleaning

03

```
df_order.info() # shipping jadiin time
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 112650 entries, 0 to 112649
Data columns (total 15 columns):
 #   Column                         Non-Null Count   Dtype  
---  ------                         --------------   -----  
 0   order_id                       112650 non-null  object 
 1   order_item_id                  112650 non-null  int64  
 2   product_id                     112650 non-null  object 
 3   seller_id                      112650 non-null  object 
 4   shipping_limit_date            112650 non-null  object 
 5   price                          112650 non-null  float64
 6   freight_value                  112650 non-null  float64
 7   product_category_name          111047 non-null  object 
 8   product_category_name_english  111023 non-null  object 
 9   seller_zip_code_prefix         112650 non-null  int64  
 10  seller_city                    112650 non-null  object 
 11  seller_state                   112650 non-null  object 
 12  geolocation_zip_code_prefix    112397 non-null  float64
 13  geolocation_lat                112397 non-null  float64
 14  geolocation_lng                112397 non-null  float64
dtypes: float64(5), int64(2), object(8)
memory usage: 13.8+ MB
```

Terlihat bahwa pada df_order terdapat kolom dengan tipe data yang tak seharusnya yaitu shipping_limit_date

```
df_order["shipping_limit_date"] = pd.to_datetime(df_order["shipping_limit_date"])
```

Mengubah tipe data shipping_limit_date menjadi datetime

```
df_order.isna().sum() # Terdapat missing value
```

```
order_id                           0
order_item_id                      0
product_id                         0
seller_id                          0
shipping_limit_date                0
price                              0
freight_value                      0
product_category_name           1603
product_category_name_english   1627
seller_zip_code_prefix             0
seller_city                        0
seller_state                       0
geolocation_zip_code_prefix      253
geolocation_lat                  253
geolocation_lng                  253
dtype: int64
```

Memeriksa jumlah missing value dan duplikat pada df_order

```
print("Jumlah duplikasi: ",df_order.duplicated().sum())
```

```
Jumlah duplikasi:  0
```

```
df_order['product_category_name'].fillna('not defined', inplace=True)
df_order['product_category_name_english'].fillna('not defined', inplace=True)

df_order["product_category_name_english"] =
np.where(df_order["product_category_name"] == 'pc_gamer', 'PC Gaming', df_order["product_category_name_english"])
df_order["product_category_name_english"] =
np.where(df_order["product_category_name"] == 'portateis_cozinha_e_preparadores_de_alimentos', 'portable kitchen food prepa
```

Menangani missing value dengan melakukan translate nama untuk kolom product_category_name_english yang kosong.

```
order.info() # salah tipe data pada order_purchase_timestamp , order_app
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 103887 entries, 0 to 103886
Data columns (total 16 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   order_id                       103887 non-null  object
 1   customer_id                    103887 non-null  object
 2   order_status                   103887 non-null  object
 3   order_purchase_timestamp       103887 non-null  object
 4   order_approved_at              103712 non-null  object
 5   order_delivered_carrier_date   101999 non-null  object
 6   order_delivered_customer_date  100755 non-null  object
 7   order_estimated_delivery_date  103887 non-null  object
 8   payment_type                   103886 non-null  object
 9   payment_value                  103886 non-null  float64
 10  customer_zip_code_prefix       103887 non-null  int64
 11  customer_city                  103887 non-null  object
 12  customer_state                 103887 non-null  object
 13  geolocation_zip_code_prefix    103600 non-null  float64
 14  geolocation_lat                103600 non-null  float64
 15  geolocation_lng                103600 non-null  float64
dtypes: float64(4), int64(1), object(11)
memory usage: 13.5+ MB
```

Terlihat bahwa pada order terdapat kolom dengan tipe data yang tak seharusnya yaitu order_purchase_timestamp , order_approved_at, order_delivered_carrier_date, order_delivered_customer_date ,order_estimated_delivery_date dan order status

```
order['order_purchase_timestamp'] = pd.to_datetime(order['order_purchase_timestamp'])
order['order_approved_at'] = pd.to_datetime(order['order_approved_at'])
order['order_delivered_carrier_date'] = pd.to_datetime(order['order_delivered_carrier_date'])
order['order_delivered_customer_date'] = pd.to_datetime(order['order_delivered_customer_date'])
order['order_estimated_delivery_date'] = pd.to_datetime(order['order_estimated_delivery_date'])
order['order_status'] = order['order_status'].astype('category')
```

Mengubah tipe data order_purchase_timestamp , order_approved_at, order_delivered_carrier_date, order_delivered_customer_date dan order_estimated_delivery_date menjadi datetime sedangkan order status menjadi tipe data kategori.

```
order.loc[order['order_status'] == 'shipped']
```

Penambahan kategori order_status

```
order.isna().sum()
```

```
order_id                            0
customer_id                         0
order_status                        0
order_purchase_timestamp            0
order_approved_at                 175
order_delivered_carrier_date     1888
order_delivered_customer_date    3132
order_estimated_delivery_date       0
payment_type                        1
payment_value                       1
customer_zip_code_prefix            0
customer_city                       0
customer_state                      0
geolocation_zip_code_prefix       287
geolocation_lat                   287
geolocation_lng                   287
dtype: int64
```

Memeriksa jumlah missing value dan duplikat pada order

```
print("Jumlah duplikasi: ",order.duplicated().sum())
```

```
Jumlah duplikasi:  615
```

```
order = order.dropna(subset = ["payment_type","payment_value"])
```

Melakukan drop pada payment_type dan payment_value

```
order = order.drop_duplicates()
```

Melakukan drop duplikat

EDA

04

```
df_order.describe(include="all")
```

```
ime_is_numeric=True` to silence this warning and adopt the future behavior now.
```

name_english	seller_zip_code_prefix	seller_city	seller_state	geolocation_zip_code_prefix	geolocation_lat	geolocation_lng
112650	112650.000000	112650	112650	112397.000000	112397.000000	112397.000000
74	NaN	611	23	NaN	NaN	NaN
bed_bath_table	NaN	sao paulo	SP	NaN	NaN	NaN
11115	NaN	27983	80342	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	24439.170431	NaN	NaN	24435.840191	-22.800558	-47.235919
NaN	27596.030909	NaN	NaN	27593.085486	2.697063	2.341211
NaN	1001.000000	NaN	NaN	1001.000000	-36.605374	-67.809656
NaN	6429.000000	NaN	NaN	6429.000000	-23.610305	-48.831547
NaN	13568.000000	NaN	NaN	13568.000000	-23.422313	-46.747050
NaN	27930.000000	NaN	NaN	27345.000000	-21.766477	-46.518082
NaN	99730.000000	NaN	NaN	99730.000000	-2.546079	-34.847856

```
[ ] # mendefinisikan fungsi yang akan digunakan untuk EDA
    def range(series):
        return series.max() - series.min()
```

```
df_order.groupby(by="product_category_name_english").agg({
    "product_id": "count", #jumlah pembelian
    "price": ["max", "min", "mean", range]
}).sort_values(by=("product_id", "count"), ascending=False)
```

	product_id	price			
	count	max	min	mean	range
product_category_name_english					
bed_bath_table	11115	1999.98	6.99	93.296327	1992.99
health_beauty	9670	3124.00	1.20	130.163531	3122.80
sports_leisure	8641	4059.00	4.50	114.344285	4054.50
furniture_decor	8334	1899.00	4.90	87.564494	1894.10
computers_accessories	7827	3699.99	3.90	116.513903	3696.09
...	...	...	...	...	...
cds_dvds_musicals	14	65.00	45.00	52.142857	20.00
la_cuisine	14	389.00	24.00	146.785000	365.00
PC Gaming	9	239.00	129.99	171.772222	109.01
fashion_childrens_clothes	8	110.00	39.99	71.231250	70.01
security_and_services	2	183.29	100.00	141.645000	83.29

74 rows × 5 columns

Dari tabel disamping bisa terlihat bahwa nama kategori yang paling banyak dibeli adalah bed_bath_table dan kota seller yang paling menjual adalah sao paulo

bed_bath_table adalah produk yang banyak dibeli dan health_beauty adalah produk yang menghasilkan rata rata harga tertinggi

```
df_order.groupby(by="seller_state").seller_id.nunique().sort_values(ascending=False)
```

```
seller_state
SP    1849
PR     349
MG     244
SC     190
RJ     171
RS     129
GO      40
DF      30
ES      23
BA      19
CE      13
PE       9
PB       6
MS       5
RN       5
MT       4
RO       2
SE       2
AC       1
PI       1
AM       1
MA       1
PA       1
Name: seller_id, dtype: int64
```

SP merupakan state dengan penjual terbanyak

```
df_order.groupby(by="seller_city").seller_id.nunique().sort_values(ascending=False)
```

```
seller_city
sao paulo         694
curitiba          127
rio de janeiro     96
belo horizonte     68
ribeirao preto     52
                 ...
ivoti               1
itirapina           1
itau de minas       1
itapui              1
xaxim               1
Name: seller_id, Length: 611, dtype: int64
```

Sao paulo merupakan kota dengan seller terbanyak

```
order.groupby(by="customer_city").customer_id.nunique().sort_values(ascending=False)
```

```
customer_city
sao paulo         15540
rio de janeiro     6882
belo horizonte     2773
brasilia           2131
curitiba           1521
                  ...
ibiara                1
rio espera            1
rio dos indios        1
rio dos cedros        1
lagoao                1
Name: customer_id, Length: 4119, dtype: int64
```

Sao paulo merupakan kota dengan customer terbanyak

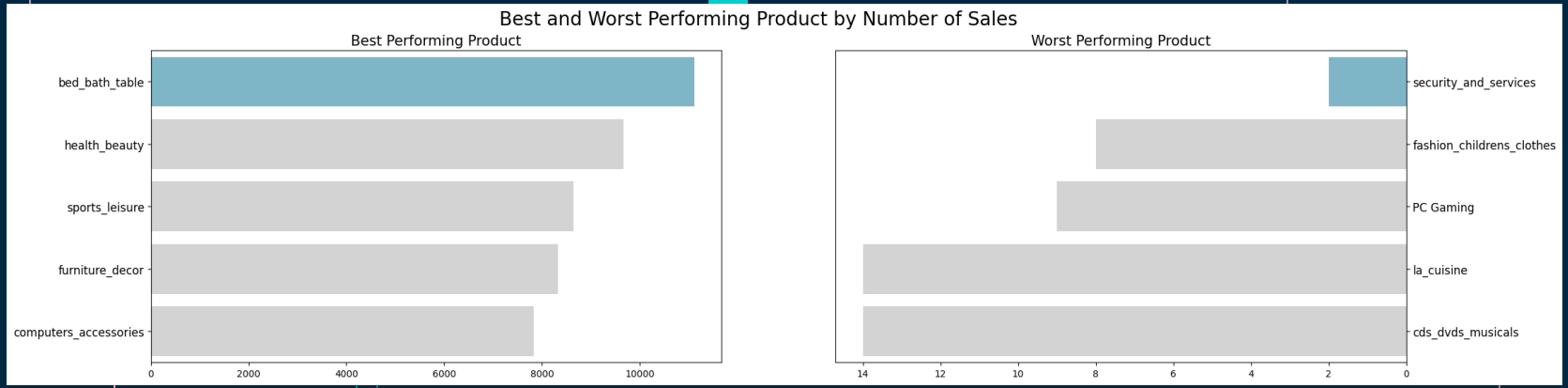
```
order.groupby(by="customer_state").customer_id.nunique().sort_values(ascending=False)
```

```
customer_state
SP    41745
RJ    12852
MG    11635
RS     5466
PR     5045
SC     3637
BA     3380
DF     2140
ES     2033
GO     2020
PE     1652
CE     1336
PA      975
MT      907
MA      747
MS      715
PB      536
PI      495
RN      485
AL      413
SE      350
TO      280
RO      253
AM      148
AC       81
AP       68
RR       46
Name: customer_id, dtype: int64
```

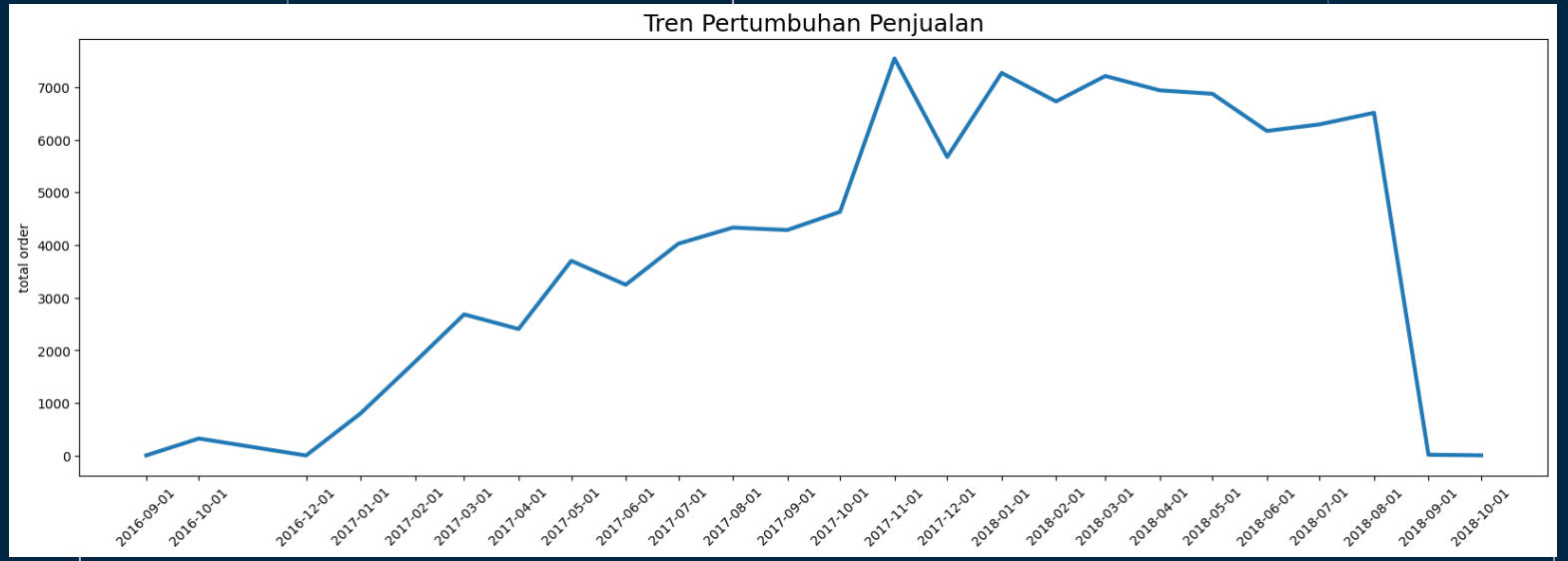
SP merupakan state dengan customer terbanyak

Visualitation & Explanatory Analysis

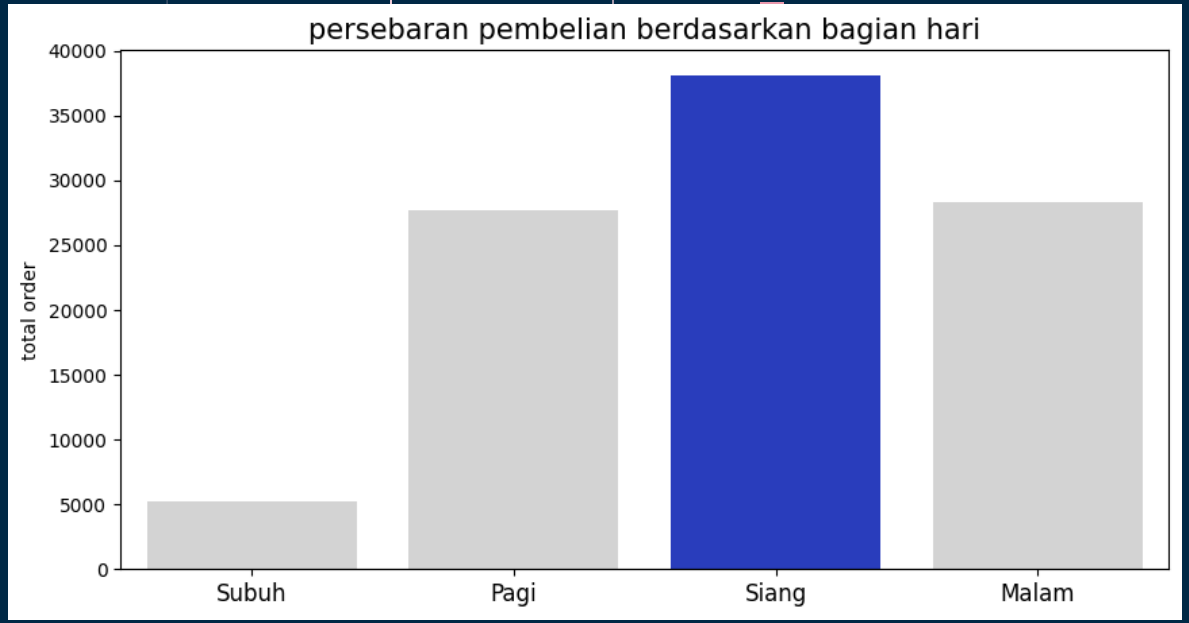
05



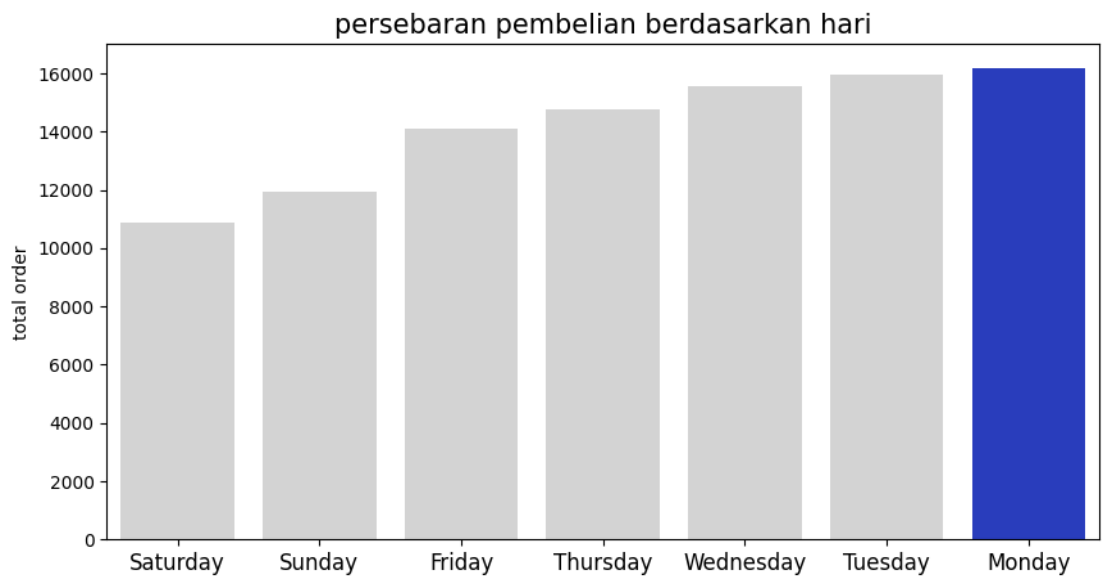
Terlihat bahwa kategori barang yang paling banyak dibeli adalah bed_bath_table sedangkan paling sedikit dibeli adalah security_and_services



Pembelian terbanyak terjadi pada bulan November tahun 2017



Pembelian terbanyak terjadi pada siang hari



Pembelian terbanyak terjadi pada hari senin

Kesimpulan

06

Kesimpulan

1. Apa kategori barang yang paling banyak dibeli dan paling sedikit diminati?

Kategori barang yang paling banyak dibeli adalah bed_bath_table sedangkan paling sedikit dibeli adalah security_and_services.

2. Bulan apa yang terjadi penjualan tertinggi?

Pembelian terbanyak terjadi pada bulan November tahun 2017.

3. Hari apa yang sering digunakan oleh pembeli untuk melakukan transaksi?

Hari Senin adalah hari yang paling banyak digunakan oleh konsumen untuk belanja dan umumnya konsumen melakukan transaksi pada siang hari.