Technical Test Answers

Nama : Jafar Shodiq

File : ipynb (terlampir)

Dataset

1. Mengkategorikan semua jkt-hub menjadi Jakarta

```
for i in df['kota']:
    if 'hub' in i:
        df['kota'] = df['kota'].str.replace(i, 'jakarta')
    if 'jakarta' in i:
        x = i[:7]
        df['kota'] = df['kota'].str.replace(i, x)

df[['kota', 'status']][df['kota'].str.contains('jakarta')].sample(10)
```

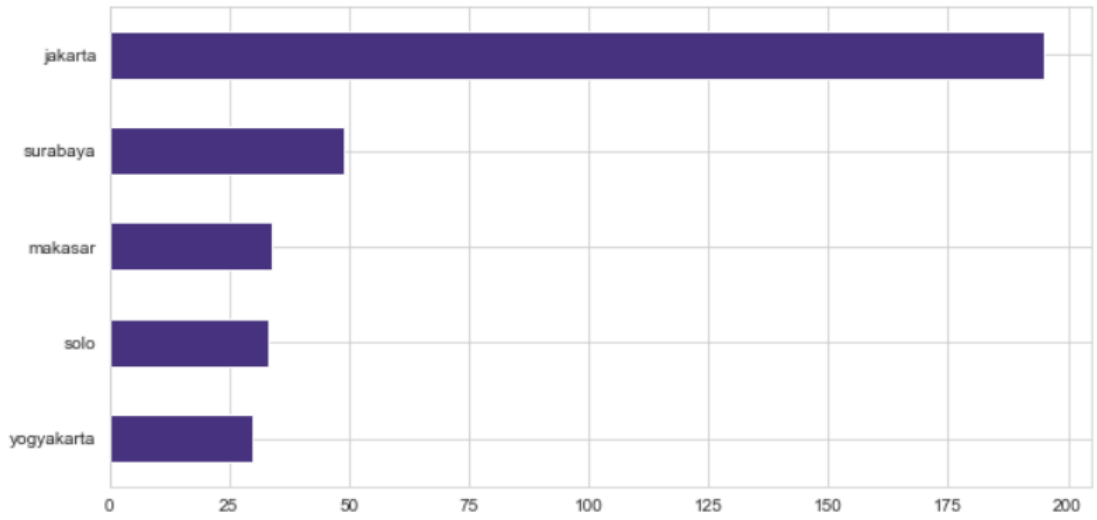
	kota	status
231	jakarta	off
354	jakarta	aktif
311	jakarta	off
428	jakarta	aktif
348	jakarta	off
889	jakarta	aktif
6	jakarta	rpc
191	jakarta	rpc
1071	jakarta	aktif
857	jakarta	hold

2. Jumlah ATM yang aktif di top 5 kota

```
df2 = df[['kota', 'status']]
df2['kota'][df2['status'].str.contains('aktif')].value_counts().head()
```

```
jakarta      195
surabaya      49
makasar       34
solo          33
yogyakarta    30
Name: kota, dtype: int64
```

```
df2['kota'][df2['status'].str.contains('aktif')].value_counts().head().sort_values().plot(kind='barh', figsize=(10, 5))
```



3. Di kota Jakarta, penyewaan ATM yang aktif ada di kategori lokasi mana saja

```
df3 = df[['lokasi']][(df['kota']=='jakarta') & (df['status']=='aktif')]
```

Me-list semua kategori yang mungkin:

- Swalayan

```
lst_swalayan = []
lst_swalayan_cat = ['toko', 'swalayan', 'tip top', 'carefour',
'indomaret', 'alfa mart', 'alfamart', 'giant', 'super indo']
for i in lst_swalayan_cat:
    for x in list(df3['lokasi'][df3['lokasi'].str.contains(i)]):
        lst_swalayan.append(x)
len_swalayan = len(list(set(lst_swalayan)))
```

- SPBU

```
lst_spbu = []
lst_spbu_cat = ['spbu']
for i in lst_spbu_cat:
    for x in list(df3['lokasi'][df3['lokasi'].str.contains(i)]):
        lst_spbu.append(x)
len_spbu = len(list(set(lst_spbu)))
```

- Kantor Cabang Pembantu (KCP)

```
lst_kcp = []
lst_kcp_cat = ['kcp']
for i in lst_kcp_cat:
    for x in list(df3['lokasi'][df3['lokasi'].str.contains(i)]):
        lst_kcp.append(x)
len_kcp = len(list(set(lst_kcp)))
```

- Mall

```
lst_mall = []
lst_mall_cat = ['mall', 'plaza']
for i in lst_mall_cat:
    for x in list(df3['lokasi'][df3['lokasi'].str.contains(i)]):
        lst_mall.append(x)
len_mall = len(list(set(lst_mall)))
```

- Universitas

```
lst_univ = []
lst_univ_cat = ['universitas', 'univ', 'kampus']
for i in lst_univ_cat:
    for x in list(df3['lokasi'][df3['lokasi'].str.contains(i)]):
        lst_univ.append(x)
len_univ = len(list(set(lst_univ)))
```

- PLN

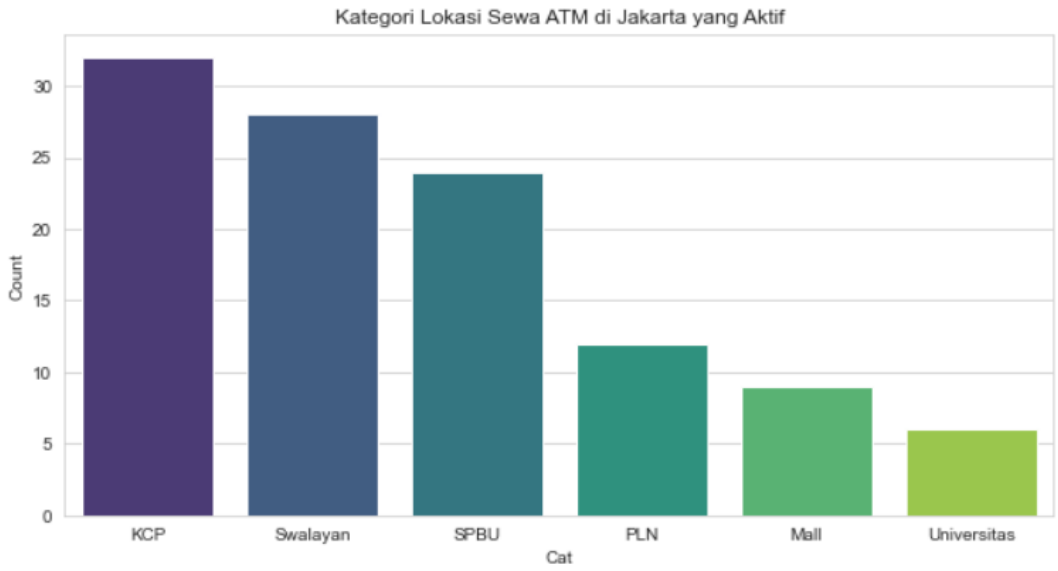
```
lst_pln = []
lst_pln_cat = ['pln']
for i in lst_pln_cat:
    for x in list(df3['lokasi'][df3['lokasi'].str.contains(i)]):
        lst_pln.append(x)
len_pln = len(list(set(lst_pln)))
```

Visualisasi:

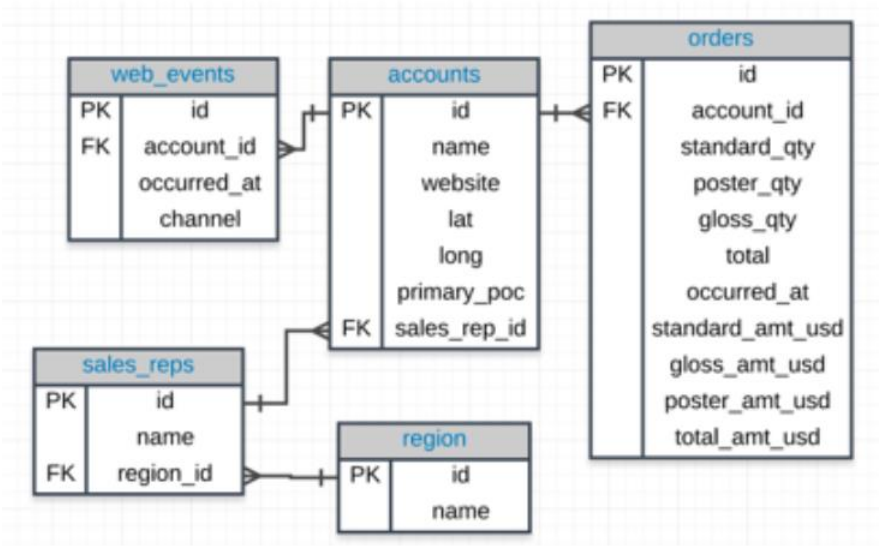
```
new_df_lokasi = pd.DataFrame(columns=('Cat', 'Count'))
cat_lst = ['Swalayan', 'Mall', 'SPBU', 'Universitas', 'KCP', 'PLN']
len_lst = [len_swalayan, len_mall, len_spbu, len_univ, len_kcp, len_pln]
new_df_lokasi['Cat'] = cat_lst
new_df_lokasi['Count'] = len_lst
new_df_lokasi.sort_values('Count', ascending=False, inplace=True)
new_df_lokasi
```

	Cat	Count
4	KCP	32
0	Swalayan	28
2	SPBU	24
5	PLN	12
1	Mall	9
3	Universitas	6

```
plt.figure(figsize=(10, 5))
sns.barplot(x='Cat', y='Count', data=new_df_lokasi)
plt.title('Kategori Lokasi Sewa ATM di Jakarta yang Aktif')
```



SQL



1. Provide the **name** for each region for every **order**, as well as the account **name** and the **unit price** they paid (total_amt_usd/total) for the order. However, you should only provide the results if the **standard order quantity** exceeds 100 and the **poster order quantity** exceeds 50. Your final table should have 3 columns: **region name**, **account name**, and **unit price**. Sort for the largest **unit price** first.

```
SELECT region.name AS region_name, accounts.name AS account_name,
(orders.total_amt_usd/orders.total) AS unit_price
FROM accounts
JOIN orders
ON orders.account_id=accounts.id
JOIN sales_reps
```

```
ON sales_reps.id=accounts.sales_rep_id
JOIN region
ON region.id=sales_reps.region_id
WHERE orders.standard_qty>100 AND orders.poster_qty>50
ORDER BY unit_price DESC
```

2. Provide the **name** of the **sales_rep** in each **region** with the largest amount of **total_amt_usd** sales.

```
SELECT sales_reps.name, MAX(orders.total_amt_usd)
FROM sales_reps
JOIN accounts
ON sales_reps.id=accounts.sales_rep_id
JOIN orders
ON orders.account_id=accounts.id
WHERE sales_reps.region_id=1
UNION
SELECT sales_reps.name, MAX(orders.total_amt_usd)
FROM sales_reps
JOIN accounts
ON sales_reps.id=accounts.sales_rep_id
JOIN orders
ON orders.account_id=accounts.id
WHERE sales_reps.region_id=2
```

3. Find all the orders that occurred in 2015. Your final table should have 4 columns: **occurred_at**, **account name**, **order total**, and **order total_amt_usd**.

```
SELECT orders.occured_at, accounts.name, orders.total,
orders.total_amt_usd
FROM orders
JOIN accounts
ON orders.account_id=accounts.id
WHERE orders.occured_at=2015
```