Rakamin Academy

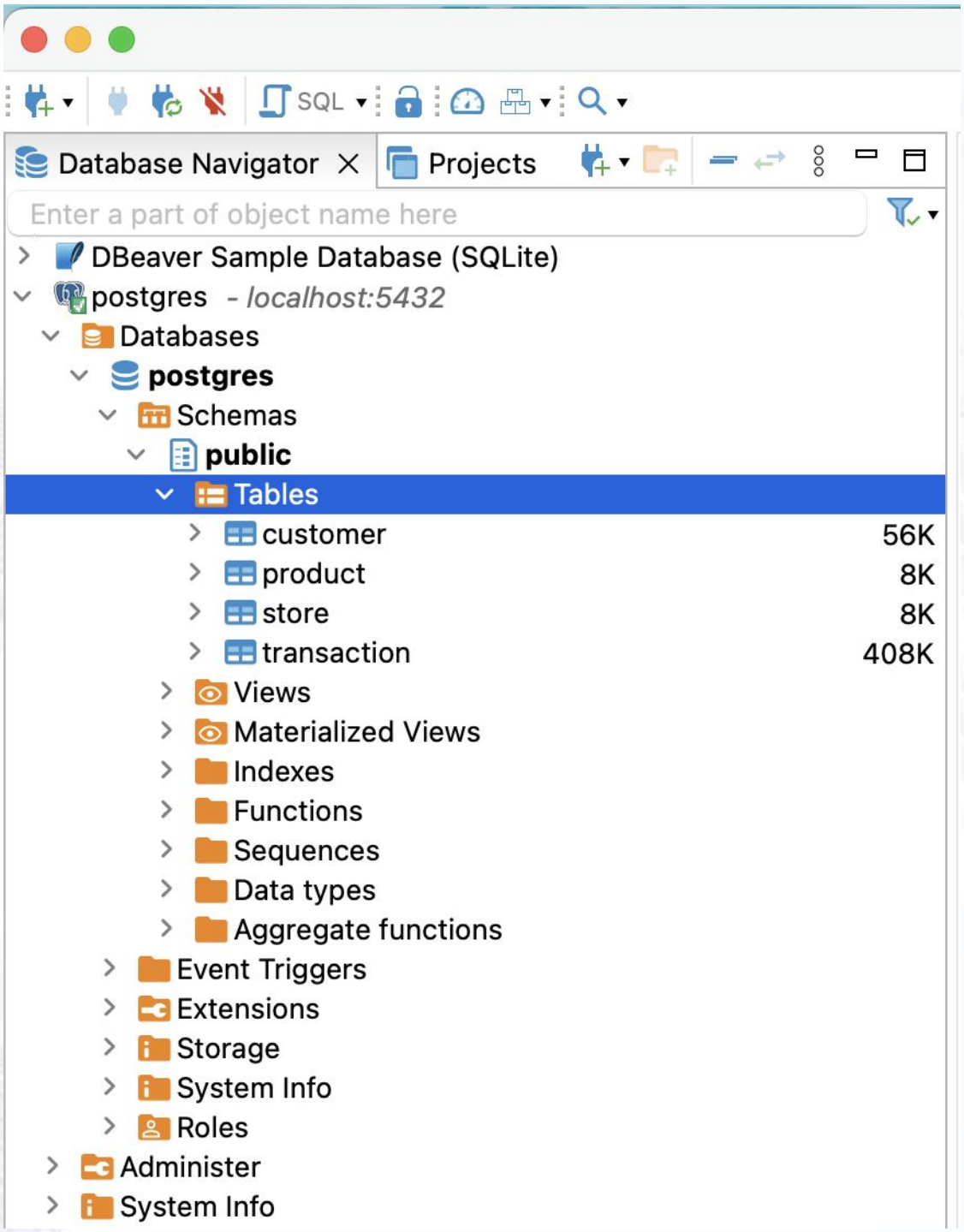
KALBE Nutritionals

Virtual Internship Experience

Machine Learning Project

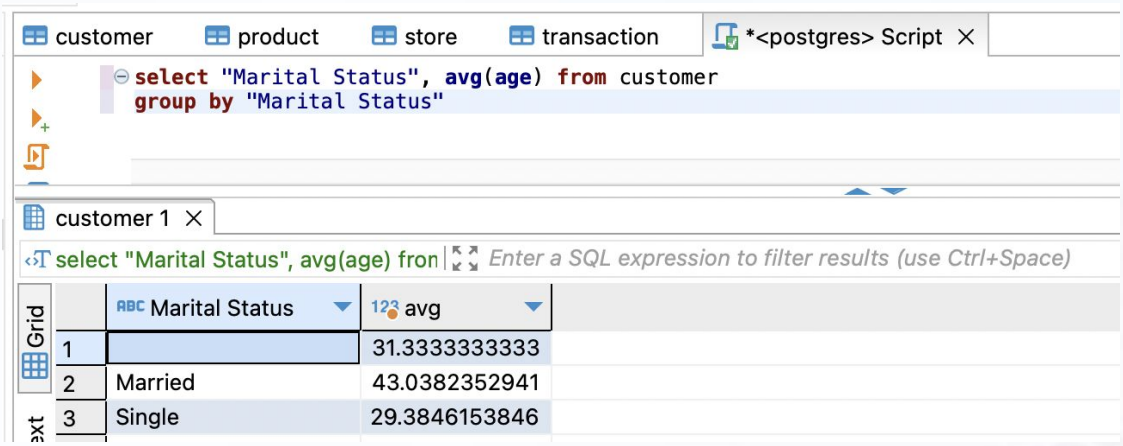
Membuat model Regression dan Clustering

Peserta dapat melakukan data ingestion ke dalam dbeaver

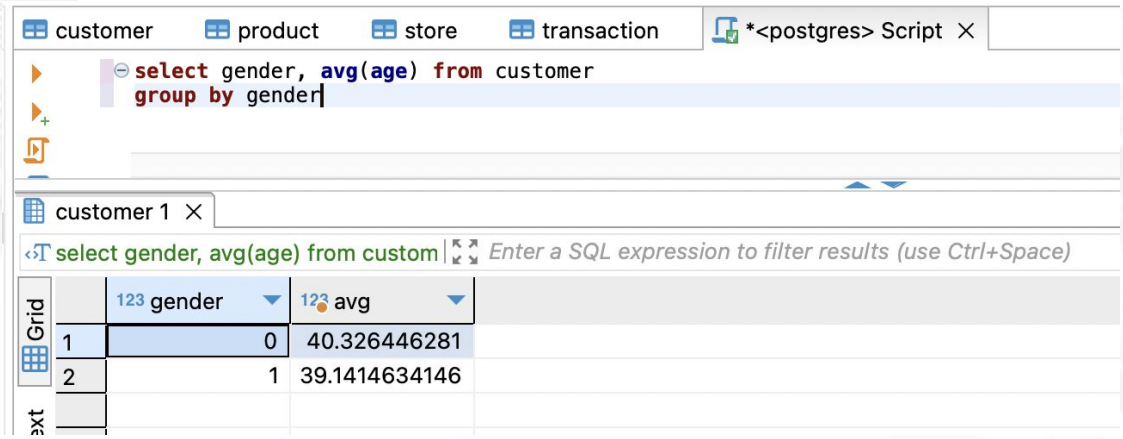


Peserta dapat melakukan exploratory data analysis di dbeaver

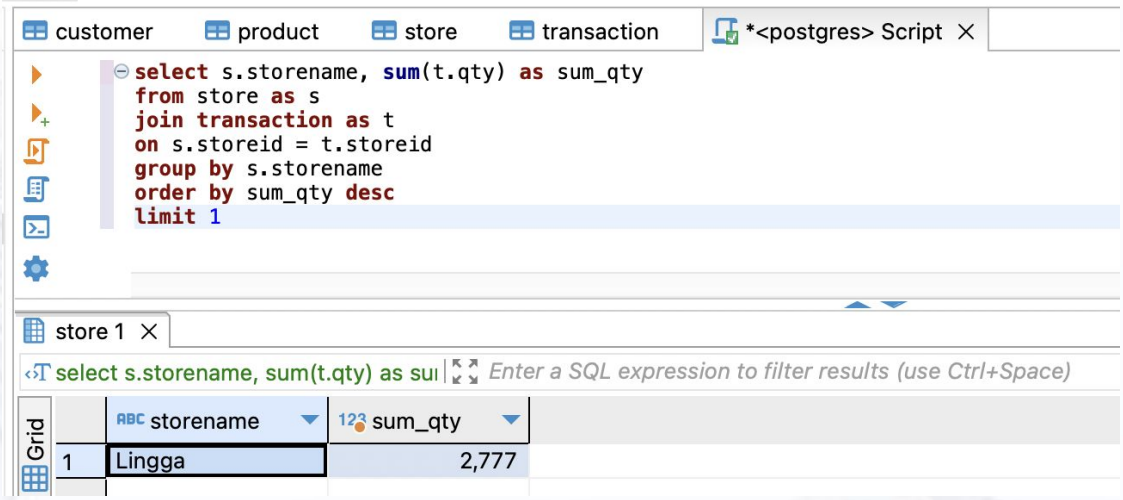
Query 1 Result



Query 2 Result



Query 3 Result



Peserta dapat melakukan exploratory data analysis di dbeaver

KALBE Nutritionals

Query 4 Result

customer | product | store | transaction | *<postgres> Script

```
select p."Product Name" , sum(t.totalamount) as sum_amount
from product as p
join transaction as t
on p.productid  = t.productid
group by p."Product Name"
order by sum_amount desc
limit 1
```

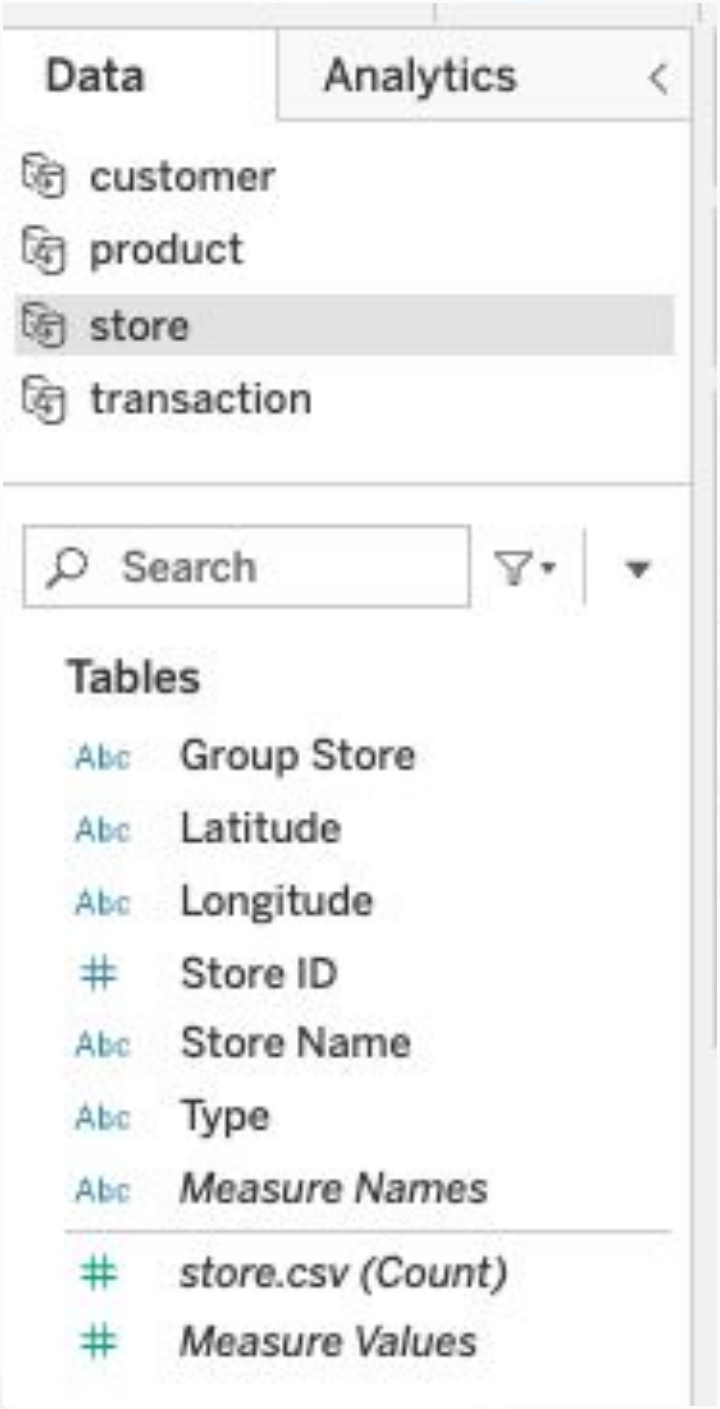
product 1

select p."Product Name" , sum(t.tota | Enter a SQL expression to filter results (use Ctrl+Space)

Grid

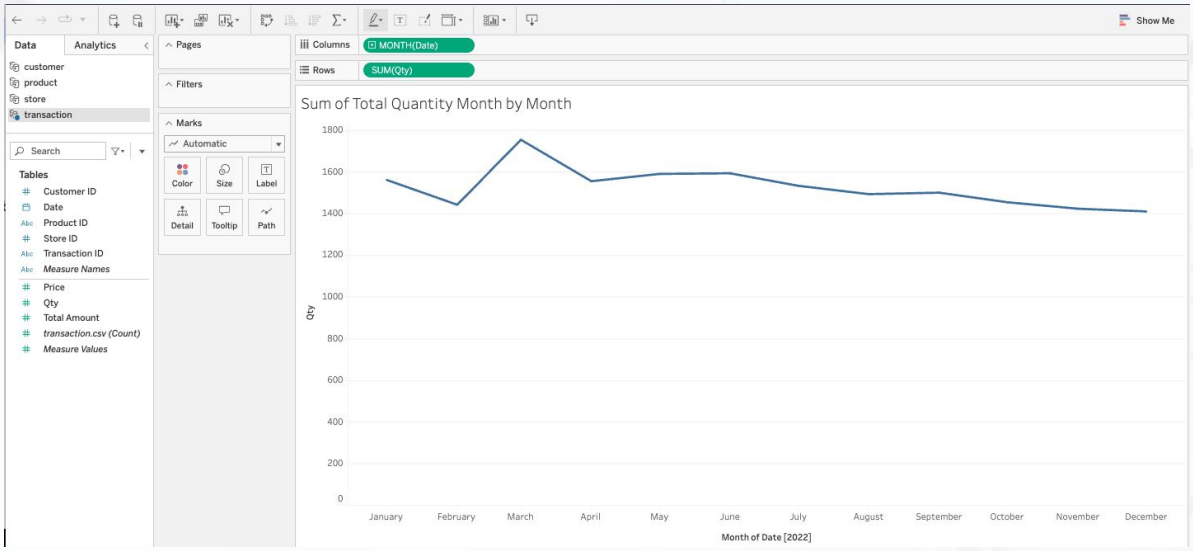
	Product Name	sum_amount
1	Cheese Stick	27,615,000

Peserta dapat melakukan data ingestion ke dalam tableau public

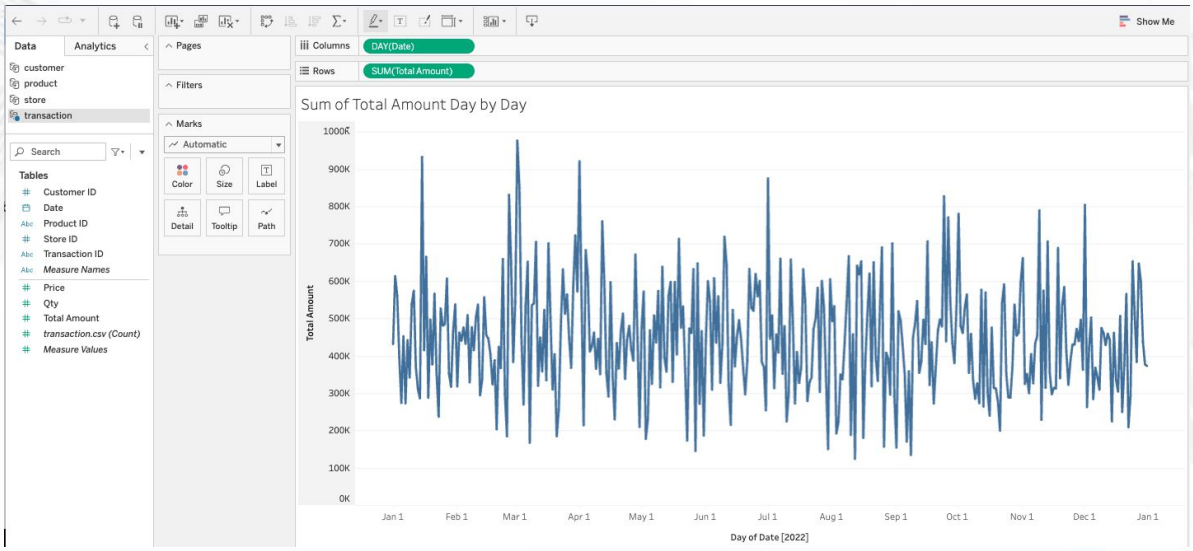


Peserta dapat membuat dashboard di tableau

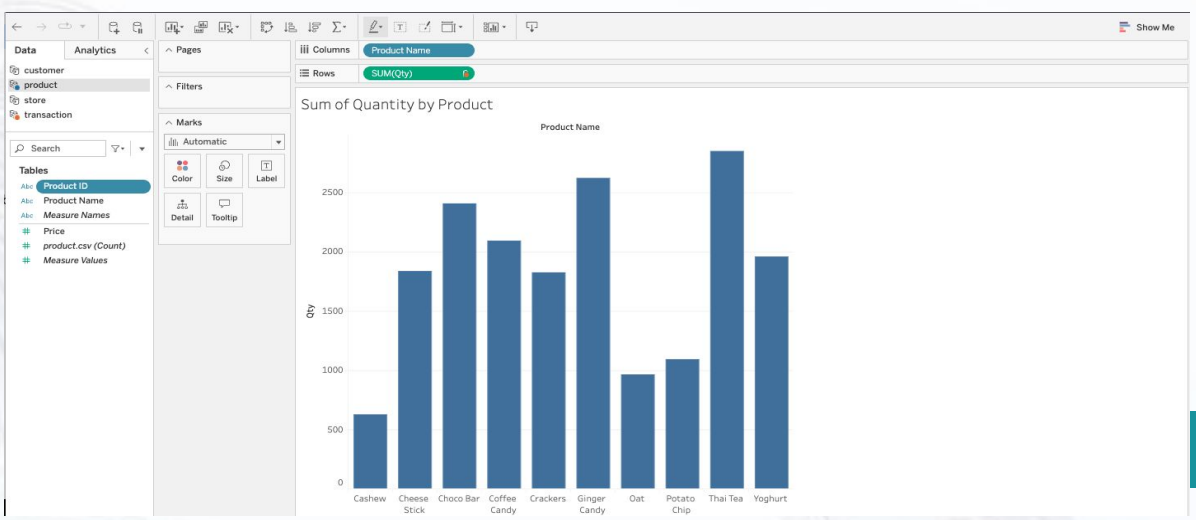
Worksheet 1 Result



Wroksheet 2 Result

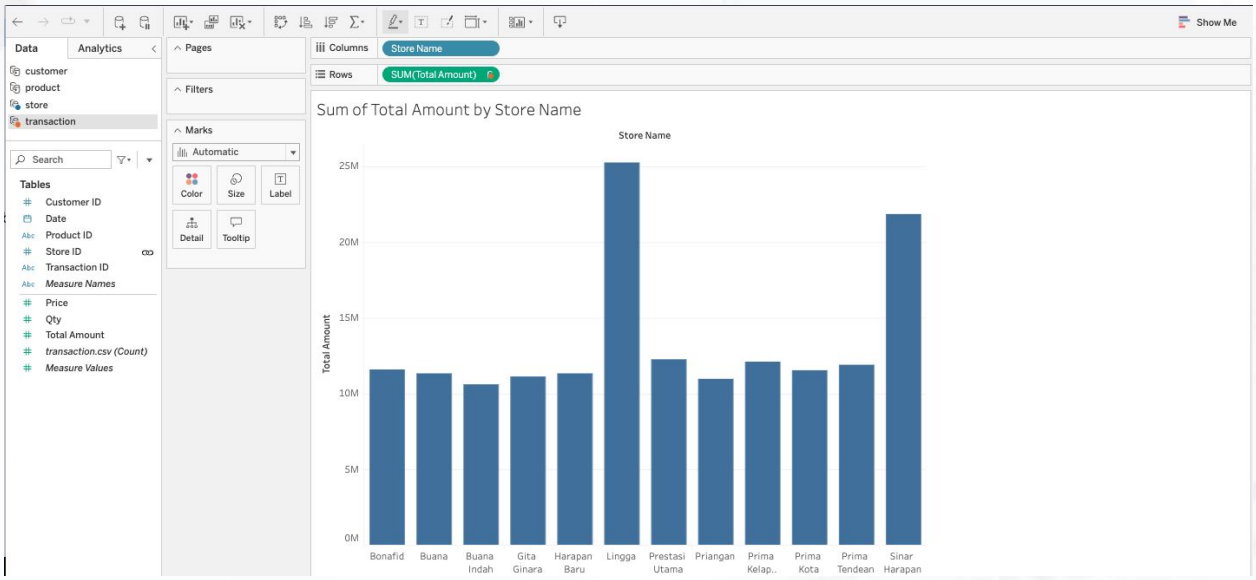


Worksheet 3 Result

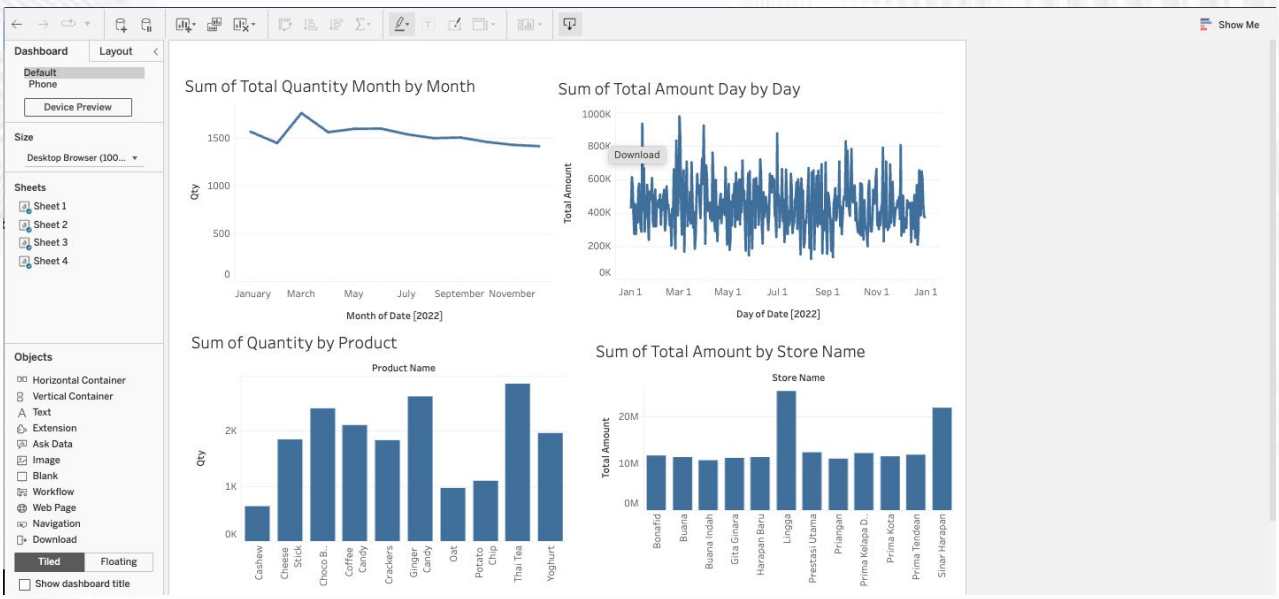


Peserta dapat membuat dashboard di tableau

Worksheet 4 Result



Dashboard Result



Peserta dapat membuat model prediktif menggunakan regresi dan membuat clustering

KALBE Nutritionals

Data Cleansing

```
[24]:
#data cleansing df customer
df_customer['Income'] = df_customer['Income'].replace('[,]','.',regex=True).astype('float')

[25]:
#data cleansing df store
df_store['Latitude'] = df_store['Latitude'].replace('[,]','.',regex=True).astype('float')
df_store['Longitude'] = df_store['Longitude'].replace('[,]','.',regex=True).astype('float')

[39]:
#data cleansing df transaction
df_transaction['Date'] = pd.to_datetime(df_transaction['Date'])
```

Data Merge

```
[40]:
df_merge = pd.merge(df_transaction, df_customer, on=['CustomerID'])
df_merge = pd.merge(df_merge, df_product.drop(columns=['Price']), on=['ProductID'])
df_merge = pd.merge(df_merge, df_store, on=['StoreID'])

[41]:
df_merge.head()
```

[41]:

	TransactionID	CustomerID	Date	ProductID	Price	Qty	TotalAmount	StoreID	Age	Gender	Marital Status	Income	Product Name	StoreName	GroupStore	Type	Latitude	Longitude
0	TR11369	328	2022-01-01	P3	7500	4	30000	12	36	0	Married	10.53	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554
1	TR89318	183	2022-07-17	P3	7500	1	7500	12	27	1	Single	0.18	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554
2	TR9106	123	2022-09-26	P3	7500	4	30000	12	34	0	Married	4.36	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554
3	TR4331	335	2022-08-01	P3	7500	3	22500	12	29	1	Single	4.74	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554
4	TR6445	181	2022-10-01	P3	7500	4	30000	12	33	1	Married	9.94	Crackers	Prestasi Utama	Prestasi	General Trade	-2.990934	104.756554

Pembuatan dataframe regresi

```
[55]:
df_regresi = df_merge.groupby(['Date']).agg({
    'Qty' : 'sum'
}).reset_index()

[56]:
df_regresi
```

[56]:

	Date	Qty
0	2022-01-01	49
1	2022-01-02	50
2	2022-01-03	76
3	2022-01-04	98
4	2022-01-05	67
...	...	...
360	2022-12-27	70
361	2022-12-28	68
362	2022-12-29	42
363	2022-12-30	44
364	2022-12-31	37

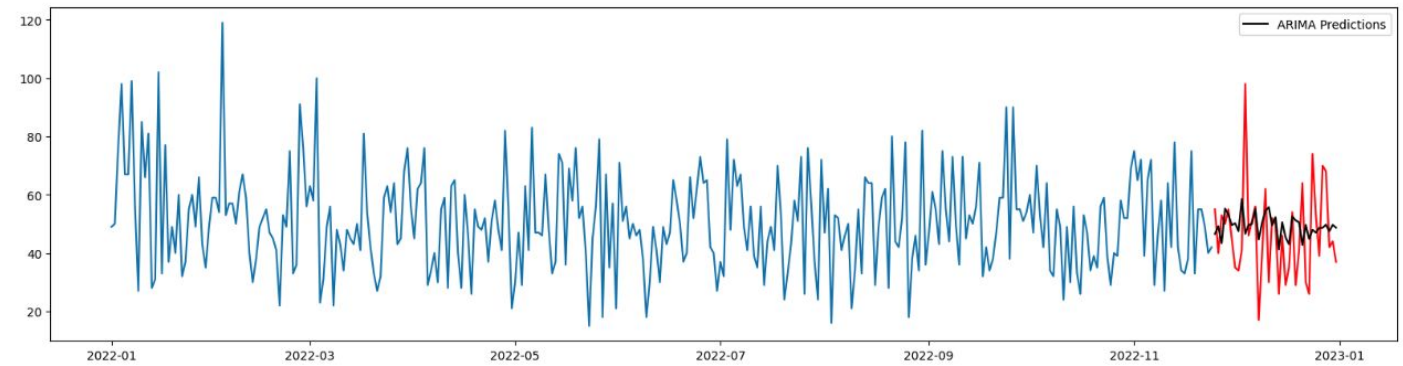
365 rows × 2 columns

Peserta dapat membuat model prediktif menggunakan regresi dan membuat clustering

Pembuatan Machine Learning ARIMA

```
[77]:
#ARIMA
df_train = df_train.set_index('Date')
df_test = df_test.set_index('Date')

y = df_train['Qty']

ARIMAmodel = ARIMA(y, order = (40, 2, 1))
ARIMAmodel = ARIMAmodel.fit()

y_pred = ARIMAmodel.get_forecast(len(df_test))

y_pred_df = y_pred.conf_int()
y_pred_df['predictions'] = ARIMAmodel.predict(start =y_pred_df.index[0], end =y_pred_df.index[-1])
y_pred_df.index = df_test.index
y_pred_out = y_pred_df['predictions']
eval(df_test['Qty'], y_pred_out)

plt.figure(figsize=(20,5))
plt.plot(df_train['Qty'])
plt.plot(df_test['Qty'], color='red')
plt.plot(y_pred_out, color='black', label = 'ARIMA Predictions')
plt.legend()
```

```
RMSE value 15.989343037134772
MAE value 12.440108877151243
```



Pembuatan dataframe clustering

```
[98]:
df_cluster = df_merge.groupby(['CustomerID']).agg({
    'TransactionID' : 'count',
    'Qty' : 'sum',
    'TotalAmount' : 'sum'
}).reset_index()
```

```
[99]:
df_cluster.head()
```

[99]:

	CustomerID	TransactionID	Qty	TotalAmount
0	1	17	60	623300
1	2	13	57	392300
2	3	15	56	446200
3	4	10	46	302500
4	5	7	27	268600

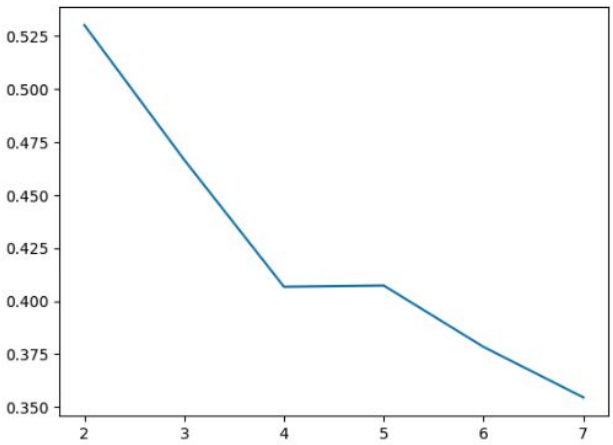
Peserta dapat membuat model prediktif menggunakan regresi dan membuat clustering

KALBE Nutritionals

Pemilihan jumlah cluster

[111]:
```
K = range(2, 8)
fits = []
score = []

for k in K:
    model = KMeans(n_clusters = k, random_state = 0, n_init='auto').fit(data_cluster_normalize)

    fits.append(model)

    score.append(silhouette_score(data_cluster_normalize, model.labels_, metric='euclidean'))
```

[112]:
```
#choose 4 cluster
sns.lineplot(x = K, y = score);
```



Analisa Cluster Final

[117]:
```
df_cluster['cluster_label'] = fits[2].labels_
```

[119]:
```
df_cluster.groupby(['cluster_label']).agg({
    'CustomerID' : 'count',
    'TransactionID' : 'mean',
    'Qty' : 'mean',
    'TotalAmount' : 'mean'
})
```

[119]:

cluster_label	CustomerID	TransactionID	Qty	TotalAmount
0	128	11.601562	40.007812	418542.187500
1	28	9.250000	35.142857	225110.714286
2	156	11.628205	42.775641	383731.410256
3	135	10.829630	40.874074	313365.925926

Thank You!

KALBE Nutritionals