LAPORAN AKHIR PROYEK PENAMBANGAN DATA

Product Clustering Using K-Means Algorithm

Disusun oleh:

1. 12S17011 Astri Monica Sianturi
2. 12S17013 Mega Sari Pasaribu
3. 12S17046 Pebri Sangmajadi Sinaga

PROGRAM STUDI SARJANA SISTEM INFORMASI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

2020

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

Bab 1. Business Understanding

Tahap pertama dari proses CRISP-DM adalah memahami apa yang ingin dicapai dari perspektif bisnis atau penelitian secara keseluruhan. Dalam tahap ini tujuan dan batasan bisnis ataupun penelitian, rencana proyek untuk mencapai data mining dan tujuan proyek, dan bagaimana strategi awal yang diperlukan serta bagaimana rancangan yang akan dibangun untuk mencapai tujuan.

Dalam kehidupan sehari-hari kita masih sering menemukan berbagai toko produk kebutuhan sehari-hari. *Retailer* akan mempromosikan produk dan layanan mereka melalui toko-toko tersebut. Proses transaksi yang berlangsung termasuk proses pemesanan, pengiriman, faktur dan pembayaran. Banyaknya persediaan jenis produk yang dapat dijual di pasar akan disebarkan ke toko-toko. Konsumen dapat melihat banyak produk baru dari waktu ke waktu dan ketertarikan konsumen akan produk juga akan berubah-ubah dari waktu ke waktu pula. Dengan berbagai pilihan dan ketertarikan konsumen terhadap produk yang dibeli menjadi salah satu tantangan kepada para penjual atau pemilik toko dalam memilih produk mana yang sebaiknya disediakan dalam waktu tertentu sesuai dengan penjualan/minat konsumen.

Penjualan produk terbanyak dapat kita ketahui dengan mengelompokkan produk-produk dari hasil transaksi yang terjadi, hasil pengelompokan tersebut akan menghasilkan beberapa kategori produk yang berbeda-beda dan kita dapat melihat jumlah produk dalam masing-masing *cluster* (kategori produk) yang terbentuk. Pentingnya mengetahui produk-produk apa saja yang memiliki angka penjualan terbanyak menjadi dasar bagi penjual dalam membuat keputusan untuk melakukan manajemen ketersediaan stok barang pada waktu yang akan datang.

Dengan demikian, tersedianya model klaster dapat membantu dalam mengkategorikan produk secara otomatis. Pengkategorian produk toko dapat dilakukan dengan menggunakan model pembelajaran *unsupervised*. Model pembelajaran *unsupervised* ditujukan untuk mengelompokan berdasarkan hasil pembelajaran komputer dalam mempelajari pola atau struktur data yang tidak ditentukan.

Clustering adalah mengelompokkan sekumpulan objek dimana objek dalam *cluster* yang sama memiliki kemiripan yang maksimal antara objek yang satu dengan yang lain daripada objek yang ada dalam cluster yang berbeda, objek yang memiliki kemiripan akan dikelompokkan dalam *cluster* yang sama. *Clustering* dapat dilakukan dengan berbagai algoritma yang berbeda

salah satu algoritma *clustering* adalah K-Means. K-Means adalah salah satu algoritma *unsupervised learning* yang paling sederhana yang memecahkan masalah *clustering* dari berbagai studi.

Beberapa penelitian terkait sudah dilakukan. Penelitian Norsyela Muhammad Noor Mathivanan dkk telah melakukan penelitian analisis *cluster* menggunakan K-Means clustering untuk mengelompokkan produk *e-commerce* dari situs toko online Malaysia. Adapun hasil penelitian yang mereka lakukan adalah terdapat tiga cluster kategori produk yang dihasilkan yaitu kategori perawatan *hair and face*, kategori perawatan *oral* dan kategori perawatan *pets*. Sehingga penelitian ini dapat menyimpulkan bahwa analisis pengelompokan menggunakan K-Means mampu mengelompokkan sekumpulan data besar secara efektif. [1]

Darmi, Yulia dan Agus Setiawan melakukan penelitian untuk menciptakan sistem yang dapat mengelompokkan produk laku dan tidak laku, yang dilakukan di Minimarket MM.TIKA Bengkulu dan dilaksanakan pada bulan Juni sampai Juli 2015. Algoritma K-Means tidak terpengaruh terhadap urutan objek yang digunakan, hal ini dibuktikan ketika penulis mencoba menentukan secara acak titik awal pusat cluster dari salah satu objek pada permulaan perhitungan. Jumlah keanggotaan cluster yang dihasilkan berjumlah sama ketika menggunakan objek yang lain sebagai titik awal pusat cluster tersebut. Namun, hal ini hanya berpengaruh pada jumlah iterasi yang dilakukan. [2]

Berdasarkan analisis permasalahan diatas, maka tim akan melakukan pengelompokan terhadap produk pada sebuah dataset hasil transaksi jual-beli. Pada proyek ini akan menghasilkan kelompok kategori produk yang dapat dimanfaatkan dalam menentukan produk yang memiliki penjualan terbanyak, algoritma yang digunakan adalah algoritma K-Means. K-Means merupakan suatu algoritma *clustering* untuk mempartisi setiap dataset hanya ke dalam satu *cluster*. Algoritma K-Means memiliki kemudahan untuk interpretasi, implementasi yang sederhana, kecepatan dalam konvergensi dan dapat beradaptasi, namun mempunyai masalah sensitivitas terhadap penentuan partisi awal jumlah *cluster*. Metode Elbow dapat dilakukan untuk memperbaiki kelemahan dari metode K-Means. Metode Elbow merupakan suatu metode yang digunakan untuk menghasilkan informasi dalam menentukan jumlah *cluster* terbaik dengan cara melihat persentase hasil perbandingan antara jumlah *cluster* yang akan membentuk siku pada suatu titik. Untuk mendapatkan perbandingannya

adalah dengan menghitung SSE (*Sum of Square Error)* dari masing-masing nilai cluster. Karena semakin besar jumlah cluster K maka nilai *SSE* akan semakin kecil. [3]

Tujuan pengerjaan proyek ini adalah menghasilkan *Product Clustering* berdasarkan data transaksi penjualan dari data transaksi toko dengan menggunakan model *Clustering (K-Means)* berdasarkan total penjualan dan harga produk. Hasil *clustering* akan digunakan dalam menganalisis penjualan produk yang dapat dimanfaatkan untuk mengambil keputusan di masa yang akan datang. Penelitian ini juga bertujuan memberikan tampilan hasil *clustering* dalam bentuk visualisasi untuk memudahkan membaca hasil *clustering*.

Bab 2. Data Understanding

Tahap selanjutnya dari proses CRISP-DM adalah memperoleh data yang akan digunakan dalam proyek untuk melakukan analisis data. Pada tahap ini juga dilakukan pemeriksaan terhadap kualitas data yang akan digunakan, apakah datanya mencakup semua kasus yang diperlukan, apakah terdapat *error* pada data, dan apakah ada *missing values* di dalam data.

2.1 Sumber Data

Pada proyek ini, data yang digunakan merupakan Store Transaction Data (*https://www.kaggle.com/iamprateek/store-transaction-data?select=Hackathon_Ideal_Data.csv*) yang diambil dari *kaggle* untuk mengidentifikasi *Product Segments* hasil transaksi penjualan produk dalam sebuah toko. Secara spesifiknya dari data ini, informasi yang ingin diperoleh adalah mengetahui kelompok kategori produk apa saja yang dibeli oleh konsumen, dan dari hal ini kita dapat mengetahui kategori produk mana yang paling banyak diminati oleh konsumen sehingga informasi tersebut dapat dimanfaatkan untuk mengambil keputusan di masa yang akan datang.

Data yang akan digunakan merupakan data penjualan top *brand level* di 10 toko dalam kurun waktu pengumpulan selama 3 bulan. Data berukuran 1.464.401 bytes ini dikemas dalam format CSV. Data terdiri dari 10 Atribut yakni Month, Storecode, Quantity (QTY), Value, Group (GRP), Subgroup (SGRP), SubSubGroup(SSGRP), Company (CMP), Mother Brand (MBRD), Brand (BRD). *Dataset* terdiri dari 14.260 baris data.

Tabel 1 Atribut pada Dataset yang Digunakan

No	Attribute	Non-Null Count	Data Type	Attribute Type
1	Month	14260 non-null	Object	Nominal
2	Storecode	14260 non-null	Object	Nominal
3	Quantity	14260 non-null	Integer	Numeric - Quantitative (Ratio Scaled)
4	Value	14260 non-null	Integer	Numeric - Quantitative (Ratio Scaled)
5	Group	14260 non-null	Object	Nominal
6	Subgroup	14260 non-null	Object	Nominal
7	SubSubGroup	14260 non-null	Object	Nominal
8	Company	14260 non-null	Object	Nominal
9.	Mother Brand	14260 non-null	Object	Nominal
10	Brand	14260 non-null	Object	Nominal

Berdasarkan sifatnya, data dapat terbagi menjadi dua jenis, yaitu data kualitatif (non-metrik) dan data kuantitatif (metrik). Data kualitatif dapat disebut data yang bukan berupa angka. Pada data kualitatif tidak bisa dilakukan operasi matematika, seperti penambahan, pengurangan, perkalian dan pembagian. Data kuantitatif dapat disebut sebagai data berupa angka. Berbagai jenis operasi matematika dapat dilakukan pada data kuantitatif. [4]

Data yang terdapat pada dataset terdiri atas data kuantitatif dan kualitatif. Data kuantitatif merupakan data yang dapat diukur (*measurable*) atau dapat dihitung sebagai angka atau bilangan. Data tersebut dapat berupa bilangan diskrit atau bilangan kontinu. Data kuantitatif memiliki kecenderungan dapat dianalisis dengan cara atau teknik statistik. Data yang termasuk kuantitatif pada *dataset* adalah Quantity (QTY) dan Value.

Data kualitatif merupakan data yang berbentuk kata, kalimat atau gambar. Data kuantitatif dapat juga disebut data kategori, yakni nominal dan ordinal. Data yang termasuk kualitatif pada dataset adalah Month, Storecode, Group (GRP), Subgroup (SGRP), SubSubGroup (SSGRP), Company (CMP), Mother Brand (MBRD), Brand (BRD).

2.3 Explore Data

Berikut merupakan hasil data explorasi yang kami lakukan menggunakan *Exploratory Data Analysis*. *Exploratory Data Analysis* (EDA) merupakan sebuah proses untuk memahami kumpulan data dengan meringkas karakteristik utamanya yang sering dilakukan dengan memplot secara visual. Langkah ini sangat penting terutama ketika kita sampai pada pemodelan data untuk penerapan *Machine Learning*.

2.3.1 Membaca Dataset

Mengimpor data dari berkas CSV akan menggunakan library *pandas* untuk membaca data dan melihat waktu eksekusinya serta melihat 5 data pertama dari dataset untuk melihat gambaran data secara umum.

```
import pandas as pd
```

```
%time data = pd.read_csv("./Hackathon_Ideal_Data.csv", delimiter=',', index_col=0)
```

```
Wall time: 332 ms
```

```
data.sample(5)
```

MONTH	STORECODE	QTY	VALUE	GRP	SGRP	SSGRP	CMP	MBRD	BRD
M1	P4	3	29	BISCUITS - CORE & NON CORE	CREAM	CREAM	MONDELEZ INTERNATIONAL	OREO	OREO VANILLA
M2	P3	6	1283	PACKAGED PURE GHEE	PACKAGED PURE GHEE	PACKAGED PURE GHEE	MOTHER DAIRY	AAREY	AAREY
M3	P4	4	40	DETERGENT CAKES/BARS	DETERGENT CAKES/BARS	DETERGENT CAKES/BARS	HARSH CLEAN DHAN PVT LTD	WOOSH	WOOSH
M3	P6	0	0	HAIR OILS PKTP(8/02)	RIGID	HAIR STYLING GEL CREAM	MARICO INDS	SET WET GELS	SET WET WET LOOK
M3	P6	21	252	SALTY SNACKS (2/97)	MIX/CHIVRA/CHANACHUR	MIX/CHIVRA/CHANACHUR	HALDIRAM	HALDIRAM	HALDIRAM PHALHARI CHIWDA

2.3.2 Melihat Kolom yang Tersedia pada Dataset

Terdapat 10 kolom pada dataset yaitu dengan instruksi data.info() maka akan ditampilkan kolom atau atribut apa saja yang terdapat dalam dataset *store transaction.*

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14260 entries, 0 to 14259
Data columns (total 10 columns):
MONTH        14260 non-null object
STORECODE    14260 non-null object
QTY          14260 non-null int64
VALUE        14260 non-null int64
GRP          14260 non-null object
SGRP         14260 non-null object
SSGRP        14260 non-null object
CMP          14260 non-null object
MBRD         14260 non-null object
BRD          14260 non-null object
dtypes: int64(2), object(8)
memory usage: 1.1+ MB
```

2.3.4 Melihat Dimensi Dataset

Melihat dimensi dataset untuk melihat berapa jumlah kolom dan baris yang terdapat pada dataset yang akan digunakan. Dimensi dari dataset yang digunakan adalah 14260 baris dan 10 kolom atau atribut.

```
data.shape
```

```
(14260, 10)
```

2.3.5 Detail Statistik

Salah satu fungsi yang terdapat pada library pandas adalah describe(). describe() digunakan untuk melihat detail statistik seperti persentil, rata-rata, standar deviasi dan lain-lain. Berikut ini adalah detail statistik dari data *store transaction* tersebut.

```
data.describe()
```

	QTY	VALUE
count	14260.000000	14260.000000
mean	16.354488	294.455330
std	34.365583	760.129558
min	0.000000	0.000000
25%	1.000000	10.000000
50%	4.000000	99.000000
75%	16.000000	283.000000
max	641.000000	24185.000000

2.3.6 Memeriksa Missing Value

Setelah kami memiliki statistik deskriptif, kemudian kami memeriksa nilai yang hilang. Pada tahap ini kami akan menampilkan 10 atribut secara berurut dengan 0% missing value untuk setiap atribut.

```
total = data.isnull().sum().sort_values(ascending=False)
percent = (data.isnull().sum()/data.isnull().count()).sort_values(ascending=False)
missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Missing Percent'])
missing_data['Missing Percent'] = missing_data['Missing Percent'].apply(lambda x: x*100)
missing_data.loc[missing_data['Missing Percent'] > 10][:10]
```

```
missing_data
```

	Total	Missing Percent
BRD	0	0.0
MBRD	0	0.0
CMP	0	0.0
SSGRP	0	0.0
SGRP	0	0.0
GRP	0	0.0
VALUE	0	0.0
QTY	0	0.0
STORECODE	0	0.0

Dari hasil keluaran di atas dapat disimpulkan bahwa dataset *store transaction* tidak memiliki *missing value.*

2.3.7 Memahami Variabel Utama

Memahami variabel utama ini merupakan bagian dari seaborn. Seaborn adalah pustaka untuk membuat grafik statistik yang menarik dan informatif dengan Python. Pustaka ini dibangun di atas ***matplotlib***, termasuk dukungan untuk struktur data ***numpy*** dan ***pandas*** serta *statistical routines* dari *scipy* dan *statsmodels*.

Mari kita membahas variabel utama yang ingin kita pahami, yaitu quantity (QTY) merupakan atribut yang menjelaskan jumlah penjualan produk. Hal pertama yang kita lakukan ketika kita memiliki variabel kategori adalah mengetahui statistik deskriptinya yaitu sebagai berikut.

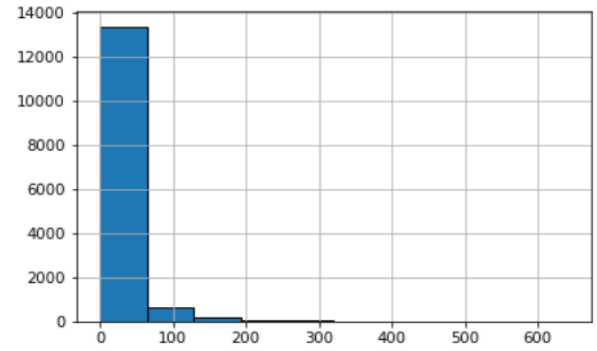
```
data['QTY'].describe()
```

```
count    14260.000000
mean        16.354488
std         34.365583
min          0.000000
25%          1.000000
50%          4.000000
75%         16.000000
max        641.000000
Name: QTY, dtype: float64
```

Dari keluaran di atas kita dapat melihat bahwa penjualan rata-rata dalam *data set* adalah 16,354488 produk untuk keseluruhan produk yang tersedia. Kita memiliki standar deviasi sekitar 34,365583. Nilai minimum penjualan produk terendah dalam *data set* adalah sekitar 0 produk, dan nilai maksimum yang sesuai dengan penjualan produk paling tinggi dalam *data set* adalah 641 produk terjual.

Memahami variabel secara visual dapat digunakan dengan cara menampilkan dalam bentuk histogram. Untuk mendapatkan histogram dari ***pandas*** Series, kita dapat menggunakan metode ***hist***, seperti yang ditunjukkan pada diagram berikut.

```
data['QTY'].hist(edgecolor = 'black', bins=10);
```



Di sini, kita melihat bahwa sangat sedikit jumlah produk terjual di atas 50 produk, sehingga sebagian besar jumlah produk terjual ada di antara 0 dan 50 produk. Hal lain yang kita perhatikan di sini adalah kita hanya memiliki sedikit data observasi dengan jumlah penjualan produk yang sangat tinggi.

2.3.8 Memahami Variabel Numerikal

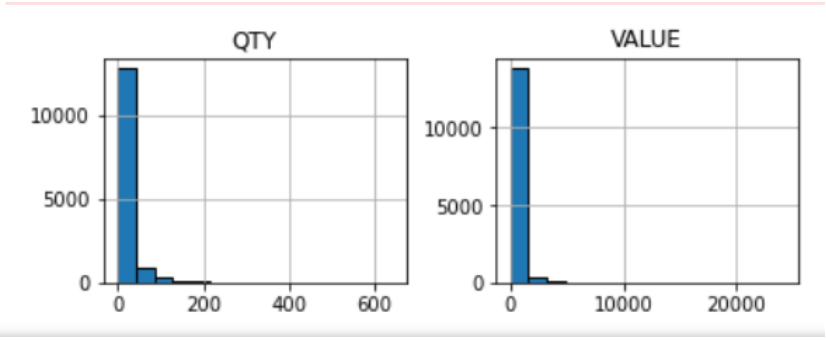
Pada tahap ini kami ingin memeriksa semua variabel numerik dalam *data set*. Untuk melakukannya, dapat dengan cara mensubstitusikan DataFrame hanya dengan mengidentifikasi variabel numerik yang ada pada data. Kita menggunakan metode describe, yang akan menampilkan sedikit DataFrame yang berisi semua statistik deskriptif untuk setiap variabel numerik yang ada di *data set*.

```
data[numerical_vars].describe()
```

	QTY	VALUE
count	14260.000000	14260.000000
mean	16.354488	294.455330
std	34.365583	760.129558
min	0.000000	0.000000
25%	1.000000	10.000000
50%	4.000000	99.000000
75%	16.000000	283.000000
max	641.000000	24185.000000

Kemudian, jika kita ingin memvisualisasikan semua variabel numerik tersebut satu per satu, dapat menerapkan metode hist ke DataFrame dan tidak hanya ke Series. Secara total akan ditampilkan 2 histogram sebagai berikut.

```
data[numerical_vars].hist(edgecolor='black', bins=15, figsize=(14, 5), layout = (2,4));
```

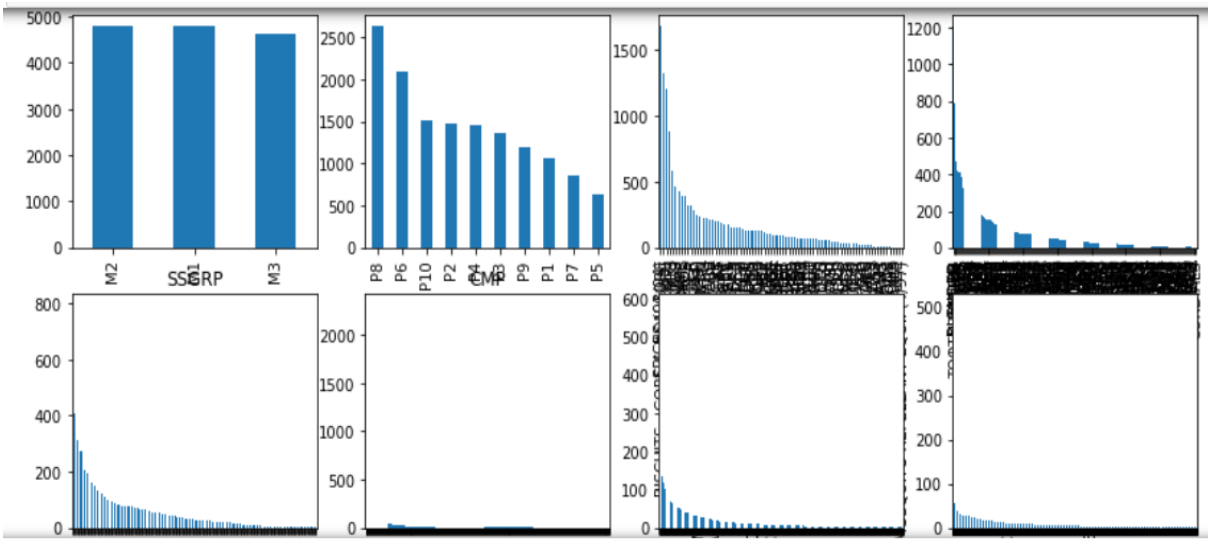


2.3.9 Memahami Variabel Kategorial

Untuk setiap variabel kategorikal, kami akan menggunakan obyek serial ***pandas*** yang akan mengkalkulasikan jumlah nilai yang ada. Kemudian akan menggunakan metode ***plot*** dengan bentuk batang, dan modifikasi yang dimiliki hanyalah menggunakan *pandas* untuk membuat plot dalam bentuk batang didalam obyek *subplot* dan variabel yang berulang. Sehingga kami akan menggunakan metode *tight_layout pad subplot* agar diagram ditampilkan dengan baik. Jumlah diagram yang dihasilkan adalah sebanyak 8 diagram untuk masing-masing variabel kategorikal yang terdapat dalam dataset.

```
fig, ax = plt.subplots(2,4, figsize=(14,6))
for var, subplot in zip(categorical_vars, ax.flatten()):
    data[var].value_counts().plot(kind='bar', ax=subplot, title=var)

fig.tight_layout()
```

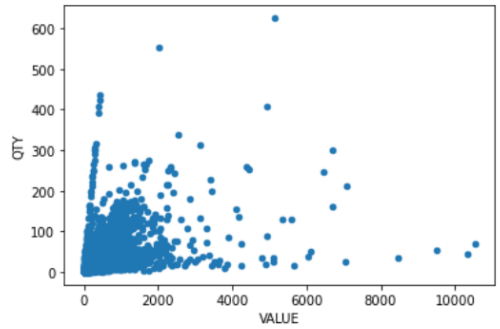


2.3.10 Memahami Hubungan Antara Variabel dengan Scatter Plot

Hubungan antara variabel yang berbeda dapat divisualisasikan dengan menggunakan *matplotlib. Plot* dengan kondisi yang kompleks akan digunakan untuk memvisualisasikan variabel pada visualisasi tersendiri. Untuk menghasilkan *scatter plot* dengan *pandas*, maka kita

akan menggunakan *plot namespace*. Pada *plot namespace*, kami menggunakan metode scatter() serta nilai x yaitu atribut *value* dan y yaitu atribut *quantity*.

```
data.plot.scatter(x='VALUE', y='QTY');
```



Dari gambar di atas kami menyimpulkan hubungan antara atribut value dengan quantity. Jadi variabel value akan dipetakan pada sumbu x dan quantity pada sumbu y, dan disini kami dapat melihat dengan jelas ada hubungan negatif antara dua variabel tersebut, semakin tinggi value produk yang dimiliki maka semakin kecil jumlah produk yang terjual, dengan kata lain semakin mahal harga produk maka semakin sedikit produk terjual.

Bab 3. Data Preparation

3.1 Data Preparation

Tahap persiapan data mencakup semua aktivitas untuk menyusun dataset akhir dari data mentah awal.

3.1.1 Data Cleaning

Cleaning yang dilakukan adalah dengan mengecek adanya nilai null, adanya data yang duplikat, dan mendeteksi adanya outlier di dalam dataset. Data Cleaning juga masih termasuk kedalam bagian dari Exploratory Data Analysis. Pada hasil *exploratory data analysis* yang dihasilkan bahwa tidak terdapat *missing value* pada dataset tersebut.

3.1.1.1 Feature Selection

Feature selection dilakukan untuk membuang sebagian kolom atau atribut yang dianggap tidak terlalu dibutuhkan, dalam proses penambangan data, dengan menggunakan fungsi drop untuk men-drop beberapa atribut.

```
data.drop(["MONTH", "STORECODE", "GRP", "SGRP",
           "SSGRP","CMP", "MBRD" ], axis = 1, inplace=True)
```

Sehingga tersisa 3 atribut yang akan digunakan yaitu QTY, VALUE, BRD.

3.1.1.1 Melakukan analisis Statistik

1. Data Statistik

	VALUE	QTY
count	14260.000000	14260.000000
mean	294.455330	16.354488
std	760.129558	34.365583
min	0.000000	0.000000
25%	10.000000	1.000000
50%	99.000000	4.000000
75%	283.000000	16.000000
max	24185.000000	641.000000

Terdapat 2 atribut numerik dan kita melihat penjelasan mengenai data statistic yang ada pada dataset tersebut.

Bab 4. Modeling

Pada tahapan ini adalah tahap mambangun model yang akan digunakan untuk menghasilkan kelompok produk.

4.1 Select modeling technique

Pada pengerjaan proyek *clustering* ini, model yang diterapkan dalam memperoleh produk yang paling banyak terjual menggunakan teknik K-Means. Menurut Mathivanan dkk (2019) Algoritma Clustering K-Means dibangun berdasarkan jarak antar titik terhadap centroid cluster. Clustering KMeans merupakan metode partisi yang paling umum digunakan untuk mengklasifikasikan objek observasi ke dalam beberapa grup yang sudah ditentukan jumlahnya (jumlah k). Biasanya, objek observasi akan dipetakan pada titik cluster dengan pusat cluster (sentroid) terdekat. Setiap cluster terdiri dari anggota dengan nilai kecocokan terdekat dan sentroid cluster akan diperbarui dari waktu ke waktu sampai semua pengamatan telah dikelompokkan masing-masing. Algoritma KMeans telah digunakan dalam pengelompokan berbagai jenis data seperti pengenalan pola, dsb. [1]

Analisis cluster/clustering merupakan proses mempartisi sekumpulan objek data (data observasi) menjadi beberapa subset. Setiap subset adalah cluster, sehingga objek dalam cluster serupa satu sama lain, namun akan terdapat perbedaan terhadap objek di cluster lain. Kumpulan cluster yang dihasilkan dari analisis cluster dapat disebut sebagai clustering. Dalam konteks ini, metode pengelompokan yang berbeda dapat menghasilkan pengelompokan yang berbeda pada kumpulan data yang sama. Clustering/pengelompokan berguna karena dapat mengarah pada penemuan grup yang sebelumnya tidak dikenal di dalam data. Data objek disebut *similiar* apabila berada pada group cluster yang sama dan data objek akan disebut *dissimiliar* apabila berada pada group cluster yang berbeda.

Algoritma KMeans dapat dilihat pada gambar berikut: [5]

1st step:
Initialize k prototypes $(x_1, \dots, x_k)$ such that $x_j = i_l, j \epsilon \{1, \dots, k\}, l \epsilon \{1, \dots, n\}$

Each cluster c_j is associated with prototype x_j

2nd step:
Repeat

for each input vector i_l , where $l \epsilon \{1, \dots, n\}$,
do
Assign i_l to the cluster c_j, with nearest prototype x_{j*}
such as $|i_l - x_{j*}| \leq |i_j - x_j|, j \epsilon \{1, \dots, k\}$
for each cluster c_j,
where $j \epsilon \{1, \dots, k\}$,
do
Update the prototype x_j to be the centroid
of all samples
currently in c_j, so that $x_j = \sum_{i_l \epsilon c_j} {}^{i_l}/_{|c_j|}$

3rd step:
Compute the error function:
$E = \sum_{j=1}^{k} \sum_{i_l \epsilon c_j} |i_j - x_j|^2$

Repeat step 1 to 3 until E does not change significantly or elements in the cluster no longer changes

4.2 Generate test design

Sebelum membangun model klister KMeans, perlu dilakukan pembuatan prosedur atau mekanisme untuk menguji kualitas dan validitas model. Dalam model klaster untuk mengurangi tingkat kesalahan maka dataset yang digunakan dipisahkan menjadi dua bagian yaitu train dan test.

4.3 Build model

Pada proyek ini, model tool yang digunakan adalah algoritma K-Means.

4.3.1 Implementasi K-Means

Pada tahap implementasi algoritma K-Means, hal pertama yang dilakukan adalah meng-*import libraries*.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

Pada code diatas yang diimpor adalah pandas, numpy, matplotlib.pyplot, dan KMeans. Untuk meng-*import* pandas menggunakan code `import pandas sebagai pd`, untuk meng-*import* numpy menggunakan code `import numpy sebagai np`.

Atribut QTY (Quantity) dan Value merupakan atribut yang akan di-*cluster*. Analisis statistik perlu dilakukan untuk mengetahui keterkaitan antar data.

```
#statistic of the data
stld_data.describe()
```

Berdasarkan hasil analisis yang ditampilkan, terdapat cukup banyak variasi dalam besaran datanya. Karena K-Means adalah algoritma yang berbasis jarak, perbedaan besaran dapat menimbulkan masalah. Oleh karena itu, besaran setiap variabel diubah ke besaran yang sama.

Berikut adalah code yang digunakan untuk menyamakan besaran setiap variabel.

```
#standadizing the data
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(stld_data)
```

Setelah besaran diubah, kita dapat menampilkan hasil analisis statistik yang besarannya telah diubah dengan menggunakan code berikut.

```
#statistics of scaled data
pd.DataFrame(data_scaled).describe()
```

Untuk memudahkan kita dalam mengetahui informasi dari data yang telah dinormalisasi, maka kita dapat memvisualisasikan data tersebut ke dalam bentuk *scatter plot*. Berikut adalah code yang dapat digunakan.

```
#create scatter plot
plt.figure(figsize=(12,6))
plt.scatter(data_scaled[:,0], data_scaled[:,1])
plt.xlim(-1,18)
plt.ylim(-2,20)
```

Setelah itu, kita akan membuat fungsi KMeans dan menyesuaikan fungsi dengan data yang akan di-*cluster*.

Pada code berikut, kita menginisialisasi jumlah cluster sebanyak dua dan menggunakan inisialisasi secara acak dimana inisialisasi secara acak dapat menghasilkan hasil cluster yang lebih baik.

```
# defining the k means function with initialization as random
kmeans = KMeans(n_clusters=2, init='random')

# fitting the k means algorithm on scaled data
kmeans.fit(data_scaled)
```

```
# save new cluster for chart
y_km = kmeans.fit_predict(data_scaled)
```

Untuk mengevaluasi cluster yang dibentuk, maka kita dapat menghitung nilai inersia dari kluster.

```
# inertia on the fitted data
kmeans.inertia_
```

Hasil inersia yang diperoleh adalah 16985.76.

Kita dapat memvisualisasikan hasil cluster dengan menggunakan code berikut.

```
#create scatter plot
plt.figure(figsize=(12,6))
plt.scatter(data_scaled[y_km ==0,0], data_scaled[y_km == 0,1], c='#003f5c')
plt.scatter(data_scaled[y_km ==1,0], data_scaled[y_km == 1,1], c='#7a5195')

plt.xlim(-1,18)
plt.ylim(-2,20)
```

Untuk menentukan jumlah cluster yang optimal menggunakan Python kita dapat menggunakan metode Elbow. Metode Elbow merupakan salah satu metode untuk menentukan jumlah cluster yang tepat melalui proporsi hasil perbandingan antara jumlah cluster yang akan membentuk siku pada suatu titik. Jika nilai cluster pertama dengan nilai cluster kedua memberikan sudut dalam grafik atau nilai mengalami penurunan paling besar maka jumlah nilai cluster tersebut yang tepat. Untuk mendapatkan perbandingannya adalah dengan menghitung Jumlah Square Error (SSE) dari masing-masing cluster nilai. Karena semakin besar jumlah nilai cluster K, maka nilai SSE akan semakin kecil. [6]

SSE dapat dihitung dengan menggunakan rumus berikut.

$$SSE = \sum_{K=1}^{K} \sum_{X_i} |x_i - c_k|^2$$

Keterangan:

K = cluster ke-c

xi = jarak data obyek ke-i

ck = pusat cluster ke-i

Kita dapat menggunakan kurva siku (*elbow curve*) untuk menentukan jumlah cluster yang optimal menggunakan Python. Kita akan menyesuaikan beberapa model k-means dan di setiap model yang berurutan, kita akan menambah jumlah cluster. Kita akan menyimpan nilai inersia dari setiap model dan kemudian memplotnya untuk memvisualisasikan hasilnya.

```
# fitting multiple k means algorithm and storing the values in a empty list
SSE = []
for cluster in range(1,20):
    kmeans = KMeans(n_jobs = -1, n_clusters = cluster, init='random')
    kmeans.fit(data_scaled)
    SSE.append(kmeans.inertia_)

#converting the results into a dataframe and plotting them
frame = pd.DataFrame({'Cluster':range(1,20), 'SSE':SSE})
plt.figure(figsize=(12,6))
plt.plot(frame['Cluster'], frame['SSE'], marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
```

Berdasarkan hasil *curve elbow* yang ditampilkan, kita dapat memilih jumlah cluster antara 3 sampai 5. Jumlah cluster yang akan digunakan adalah 5 kemudian kita akan menyesuaikan modelnya. Kita menginisialisasi jumlah cluster sebanyak lima dan menggunakan inisialisasi secara acak.

```
# k-means using 5 clusters and random initialization
kmeans = KMeans(n_jobs = -1, n_clusters = 5, init='random')
kmeans.fit(data_scaled)
pred = kmeans.predict(data_scaled)
```

Kita dapat melihat jumlah nilai poin di setiap cluster yang dibentuk diatas.

```
frame = pd.DataFrame(data_scaled)
frame['cluster'] = pred
frame['cluster'].value_counts()
```

Kita dapat menghitung nilai inersia dari kluster yang dibentuk dari jumlah cluster sebanyak 5.

```
# inertia on the fitted data
kmeans.inertia_
```

Kita dapat membuat visualisasi dari hasil cluster yang terbentuk. Tetapi sebelumnya, kita menyimpan nilai cluster terlebih dahulu dengan menggunakan kode berikut.

```
# save new cluster for chart
y_km = kmeans.fit_predict(data_scaled)
```

Untuk membuat visualisasi dari hasil cluster yang dibuat, dapat dilakukan dengan menggunakan code berikut.

```
#create scatter plot
plt.figure(figsize=(12,6))
plt.scatter(data_scaled[y_km ==0,0], data_scaled[y_km == 0,1], c='#003f5c')
plt.scatter(data_scaled[y_km ==1,0], data_scaled[y_km == 1,1], c='#7a5195')
plt.scatter(data_scaled[y_km ==2,0], data_scaled[y_km == 2,1], c='#ef5675')
plt.scatter(data_scaled[y_km ==3,0], data_scaled[y_km == 3,1], c='#808080')
plt.scatter(data_scaled[y_km ==4,0], data_scaled[y_km == 4,1], c='#00ff00')
plt.xlim(-1,18)
plt.ylim(-2,20)
```

Bab 5. Evaluation

Langkah-langkah evaluasi sebelumnya berkaitan dengan faktor-faktor seperti kualitas dan kekuatan kluster. Tahap evaluasi menilai sejauh mana model memenuhi tujuan bisnis dan usaha untuk menentukan apakah ada beberapa alasan bisnis mengapa model ini tidak memadai. Pilihan lain evaluasi adalah untuk menguji model pada aplikasi uji secara nyata aplikasi jika kendala waktu dan anggaran memungkinkan.

Selain itu, evaluasi juga menilai hasil data mining lain yang dihasilkan. Model ini akan menampilkan hasil data mining yang selalu terkait dengan tujuan bisnis asli dan semua temuan lain yang tidak selalu terkait dengan tujuan.

Salah satu kendala dalam melakukan clustering menggunakan algoritma KMeans adalah menentukan jumlah cluster yang tepat. Karena pada clustering, jumlah cluster akan mempengaruhi biaya komputasi. Semakin besar jumlah cluster maka semakin biaya komputasi juga akan meningkat.

5.1 Evaluasi Menggunakan Inersia KMeans

Salah satu langkah yang dapat dilakukan dalam menentukan cluster yang optimal adalah dengan memplot grafik atau dikenal dengan *elbow curve*, dimana sumbu x mewakili jumlah cluster dan sumbu y mewakili matrik evaluasi (inersia). Jumlah cluster yang optimal untuk data adalah dimana nilai cluster yang mengalami penurunan nilai inersia yang konstan. Berikut adalah code untuk menampilkan nilai inersia dari sebuah cluster.

```
# inertia on the fitted data
kmeans.inertia_
```

Nilai inersia dari clustering dengan jumlah cluster sebanyak 5 adalah 6768.73.

5.2 Evaluasi Menggunakan *Silhouette Coefficient*

Silhouette Coefficient digunakan untuk melihat kualitas dan kekuatan cluster, seberapa baik atau buruknya suatu objek ditempatkan dalam suatu cluster. Kriteria subjektif pengukuran pengelompokkan berdasarkan *Silhouette Coefficient* (SC) menurut Kauffman dan Roesseeuw (1990) dapat dilihat pada tabel berikut. []-> Analisis Perbandingan Metode Elbow dan Sillhouette pada Algoritma Clustering K-Medoids dalam Pengelompokan Produksi Kerajinan Bali, JURNAL MATRIX, VOL. 9, NO. 3, NOVEMBER 2019, Dewa Ayu Indah Cahya Dewi , Dewa Ayu Kadek Pramita, 104.

Tabel 2 Kriteria pengukuran Silhouette Coefficient

Nilai SC	Kriteria
0,71 – 1,00	Struktur kuat
0,51 – 0,70	Struktur baik
0,26 – 0,50	Struktur lemah
≤ 0,25	Struktur buruk

Berikut adalah code untuk menghitung nilai *Silhouette Coefficient*.

```
from sklearn.metrics import silhouette_samples, silhouette_score
```

```
silhouette_avg = silhouette_score(data_scaled, y_km)
print("For n_clusters =", n_clusters, "The average silhouette_score is :", silhouette_avg)
```

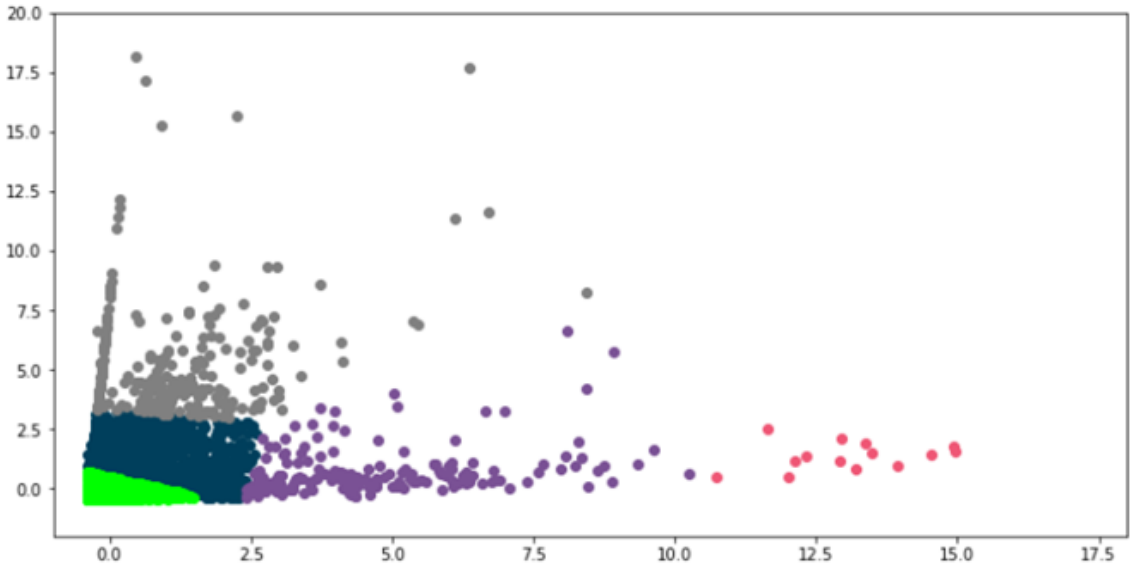
Dengan menggunakan jumlah kluster 5 maka nilai *Silhouette Coefficient* yang diperoleh adalah sebagai berikut.

```
For n_clusters = 5 The average silhouette_score is : 0.6973812680458177
```

Nilai *Silhouette Coefficient* untuk kluster 5 adalah 0,69. Berdasarkan kriteria subjektif pengukuran pengelompokkan berdasarkan *Silhouette Coefficient* (SC) menurut Kauffman dan Roesseeuw, struktur dari clustering tersebut termasuk pada kriteria yang baik.

5.3 Hasil

Hubungan antara atribut *value* dengan *quantity* dimana variabel value akan dipetakan pada sumbu x dan quantity pada sumbu y, dan disini kami dapat melihat dengan jelas ada hubungan negatif antara dua variabel tersebut, semakin tinggi value produk yang dimiliki maka semakin kecil jumlah produk yang terjual, dengan kata lain semakin mahal harga produk maka semakin sedikit produk terjual.

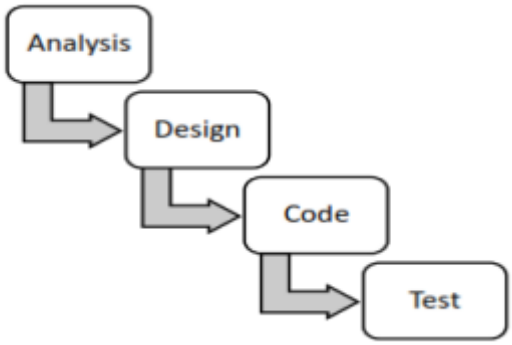


Gambar 1 Hasil Clustering

Dengan menggunakan teknik pengelompokan algoritma KMeans diperoleh kualitas dan kekuatan cluster yang sudah baik dalam proses pengelompokan data transaksi. Jumlah *cluster* yang digunakan adalah 5 *cluster*. Sehingga dapat disimpulkan bahwa penerapan *clustering* menggunakan algoritma KMeans pada data transaksi menghasilkan pengelompokan data yang baik.

6. Deployment

Tahap deployment merupakan tahap pembuatan laporan akhir atau *final report*. *Deployment* dapat juga didefinisikan sebagai proses penerapan data mining secara paralel. Saat membangun sistem, setelah evaluasi dilakukan dan hasil sesuai dengan tujuan awal, maka proses pengembangan system akan dilakukan. Namun, saat membangun sistem *product clustering*, tahap *deployment* tidak dilakukan lebih lanjut hingga menghasilkan sebuah sistem yang utuh. Metode *deployment* yang dapat digunakan adalah metode *waterfall* seperti pada berikut.



Gambar 2 Metode Waterfall

Pembangunan sistem *product clustering* diawali dari analisis (*analysis*) penentuan kebutuhan sistem, misalnya data kuantitas, value (harga) dari sebuah *brand*. Setelah analisis dilakukan, selanjutnya akan dilakukan perancanangan (*design*) untuk mempermudah pembangunan sistem. Setelah perencanaan dilakukan, pembangunan kode akan dilakukan. Pada tahap *code* dilakukan implementasi sistem yaitu penulisan kode dengan editor untuk membangun sebuah sistem serta dilakukan tahap *testing* untuk mengetahui apakah sistem yang sudah dibangun sudah sesuai dengan kebutuhan pengguna. Pada proses pembangunan sistem ini, *testing* dilakukan oleh pihak *developer*.

DAFTAR PUSTAKA

[1] N. M. N. d. Mathivanan, "Analysis of K-Means Clustering Algorithm: A Case Study Using Large Scale E-Commerce Products," *IEEE Conference on Big Data and Analytics (ICBDA),* 2019 .

[2] Y. d. A. S. Darmi, "Penerapan Metode Clustering K-Means dalam Pengelompokan Penjualan Produk," *Jurnal Media Infotama,* vol. 2, no. 2, 2016.

[3] E. d. S. K. Muningsih, "Sistem Aplikasi Berbasis Optimasi Metode Elbow untuk Penentuan Clustering Pelanggan," *JOUTICA,* vol. 3, no. 1, 2018.

[4] J. O. Ong, "Implementasi Algoritma K-Means Clustering untuk Menentukan Strategi Marketing President University," *Jurnal Ilmiah Teknik Industri,* vol. 12, no. 1, 2013.

[5] A. Dey, "Machine Learning Algorithms: A Review, International Journal of Computer Science and Information Technologies," *IEEE,* vol. Vol. 7 (3), 2016.

[6] D. A. I. C. D. A. K. P. Dewi, "Analisis Perbandingan Metode Elbow dan Sillhouette pada Algoritma Clustering K-Medoids dalam Penegelompokan Produksi Kerajinan Bali," *Jurnal Matrix,* no. 3, November 2019.