

## BAB IV

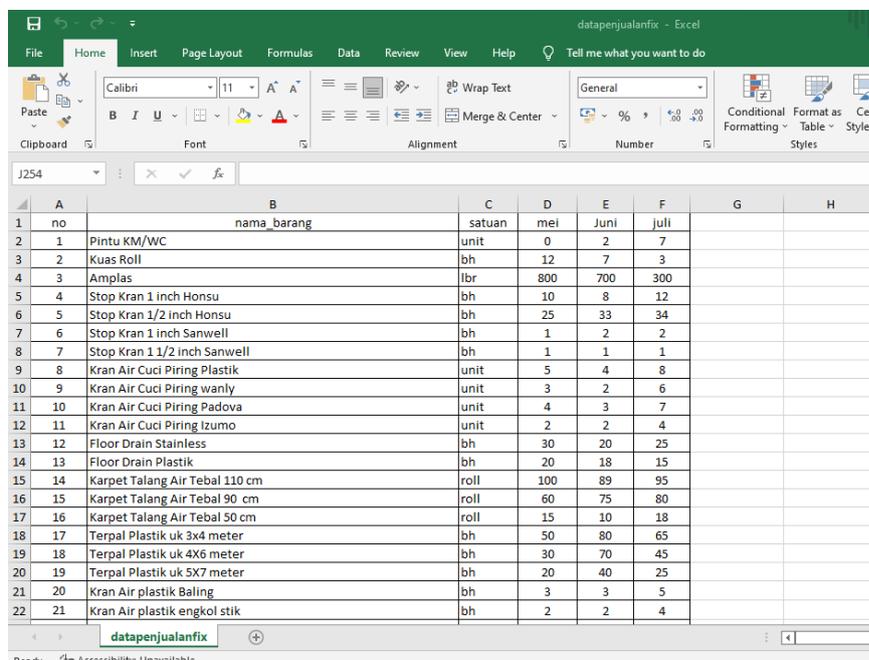
### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Hasil

##### 4.1.1 Implementasi Program

Dalam pengolahan program *Clustering* Data Penjualan pada Toko Bangunan Galunggung Untuk Mengelompokkan jenis barang yang ada pada toko, penulis mempergunakan *software* ataupun aplikasi bantu yakni aplikasi *visual studio code* serta aplikasi didalamnya yakni *Jupyter Notebook* mempergunakan Bahasa pemrograman *Python*. Program tersebut dibuat guna mengakumulasi jenis-jenis barang pada toko.

Hal awal yang dilangsungkan pengguna saat mengelolah program *Clustering K-means* yakni mempersiapkan *dataset* yang dibuat mempergunakan *Microsoft Office Excel* dengan format *.csv*.



	A	B	C	D	E	F	G	H
1	no	nama_barang	satuan	mei	Juni	juli		
2	1	Pintu KM/WC	unit	0	2	7		
3	2	Kuas Roll	bh	12	7	3		
4	3	Amplas	lbr	800	700	300		
5	4	Stop Kran 1 inch Honsu	bh	10	8	12		
6	5	Stop Kran 1/2 inch Honsu	bh	25	33	34		
7	6	Stop Kran 1 inch Sanwell	bh	1	2	2		
8	7	Stop Kran 1/2 Inch Sanwell	bh	1	1	1		
9	8	Kran Air Cuci Piring Plastik	unit	5	4	8		
10	9	Kran Air Cuci Piring wanly	unit	3	2	6		
11	10	Kran Air Cuci Piring Padova	unit	4	3	7		
12	11	Kran Air Cuci Piring Izumo	unit	2	2	4		
13	12	Floor Drain Stainless	bh	30	20	25		
14	13	Floor Drain Plastik	bh	20	18	15		
15	14	Karpet Talang Air Tebal 110 cm	roll	100	89	95		
16	15	Karpet Talang Air Tebal 90 cm	roll	60	75	80		
17	16	Karpet Talang Air Tebal 50 cm	roll	15	10	18		
18	17	Terpal Plastik uk 3x4 meter	bh	50	80	65		
19	18	Terpal Plastik uk 4x6 meter	bh	30	70	45		
20	19	Terpal Plastik uk 5x7 meter	bh	20	40	25		
21	20	Kran Air plastik Baling	bh	3	3	5		
22	21	Kran Air plastik engkol stik	bh	2	2	4		

Gambar 4.1 Dataset pada tampilan *Microsoft Excel*

File output dalam format .csv dari Microsoft Excel bakal ditampilkan seperti yang terlihat pada contoh dalam Gambar 2. Data yang disajikan merujuk pada data asli yang berasal dari Toko Bangunan Galunggung serta mencakup total 500 data barang.

Dataset yang terdiri dari 6 kolom dan berisi 500 rekaman untuk setiap kolomnya akan disubdivisikan menjadi tiga kelompok berbeda: sangat laku, laku, serta kurang laku. Sesudah proses pengelompokan selesai, langkah berikutnya yakni membuka aplikasi Visual Studio Code serta memilih alat yang disematkan di dalamnya, yakni Jupyter Notebook yang mempergunakan Bahasa pemrograman Python.

Dengan memanfaatkan berbagai perpustakaan, modul, serta paket untuk mendukung implementasi algoritma Clustering K-means serta menyediakan beragam perintah serta kode (coding) yang selaras dengan kebutuhan dan program yang ingin dijalankan oleh pengguna.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
#import library dan variabel yang digunakan untuk clustering
```

✓ 15.7s

**Gambar 4.2 Mengimport *Libarary***

Gambar 4.2 memperlihatkan proses mengimpor 6 pustaka yang bermanfaat dalam analisis data. Misalnya, numpy dipergunakan untuk kelancaran komputasi data numerik, pandas untuk pengelolaan tabel dan manipulasi data, serta seaborn untuk visualisasi data dan penyajian statistik. Selain itu, didapati pustaka lain yang digunakan untuk menerapkan teknik Clustering K-means dan menghasilkan visualisasi area plot.

```
dfbrg = pd.read_csv('datapenjualanfix.csv', sep=';', engine='python')
dfbrg.head(500)
```

✓ 0.3s

	no	nama_barang	satuan	mei	Juni	juli
0	1	Pintu KM/WC	unit	0	2	7
1	2	Kuas Roll	bh	12	7	3
2	3	Amplas	lbr	800	700	300
3	4	Stop Kran 1 inch Honsu	bh	10	8	12
4	5	Stop Kran 1/2 inch Honsu	bh	25	33	34
...	...	...	...	...	...	...
495	496	Grafel U 40 cm'	m1	28	44	52
496	497	Buis Beton 20 cm ( 1 M' )	m1	43	45	43
497	498	Buis Beton 30 cm ( 1 M' )	m1	24	35	63
498	499	Buis Beton 40 cm ( 1 M' )	m1	23	45	85
499	500	Buis Beton 50 cm ( 1 M' )	stk	23	46	25

500 rows × 6 columns

**Gambar 4.3 Bentuk DataFrame dari Dataset**

Untuk menayangkan pada bagian awal ataupun kepala dari *dataset*. Perintah diatas hanyalah menayangkan 5 data teratas dari 500 data. Sesudah itu diakibatkan penggunaberfokus pada 3 variabel yakni jumlah penjualan selama bulan “Mei”, “Juni” dan “Juli” untuk Mengelompokkan jenis-jenis barang.

Perintah ini dimaksudkan guna mengekstraksi variabel yang telah ditentukan oleh pengguna. Pada lima variabel yang tersedia, pengguna hanya memilih untuk mengekstraksi tiga variabel, yakni "mei", "Juni", dan "juli". Untuk memperlihatkan hasil dari perintah tersebut, pengguna menggunakan perintah seperti yang terlihat dalam ilustrasi:

```
dfbrg_x = dfbrg.iloc[:, 0:3]
dfbrg_x.head()
```

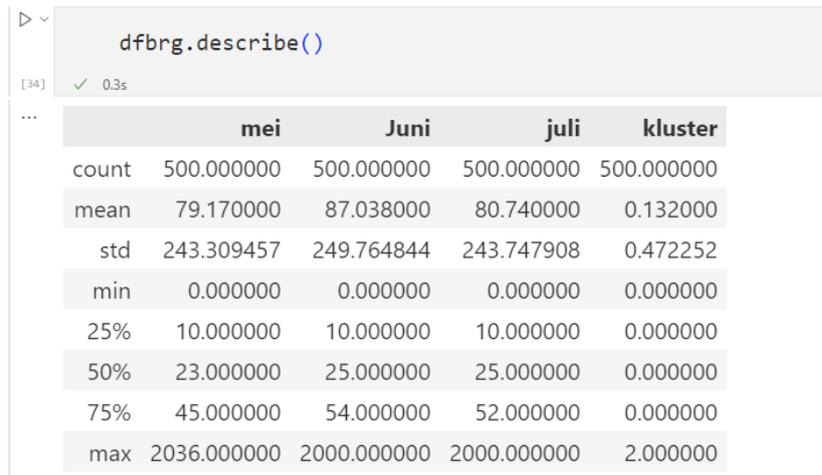
[9] ✓ 0.1s

...

	mei	Juni	juli
0	0	2	7
1	12	7	3
2	800	700	300
3	10	8	12
4	25	33	34

### Gambar 4.4 Tampilan *Dataframe*

Gambar 4.4 menayangkan 5 *dataset* teratas pada 500 *dataset*. Pada variabel yang dipergunakan juga sudah selaras dengan yang pengguna putuskan sebelumnya yakni variabel “mei”, “Juni”, dan “juli”. Sesudah itu pengguna ingin mengamati ataupun mengolah kesimpulan data statistika nya dengan melangsungkan *describe dataset*



```
dfbrg.describe()
```

	mei	Juni	juli	kluster
count	500.000000	500.000000	500.000000	500.000000
mean	79.170000	87.038000	80.740000	0.132000
std	243.309457	249.764844	243.747908	0.472252
min	0.000000	0.000000	0.000000	0.000000
25%	10.000000	10.000000	10.000000	0.000000
50%	23.000000	25.000000	25.000000	0.000000
75%	45.000000	54.000000	52.000000	0.000000
max	2036.000000	2000.000000	2000.000000	2.000000

### Gambar 4.5 *Describing Data*

Dari data visualisasinya, kita bisa melihat beberapa statistik misalnya Count, mean, std (standar deviasi), min, 25%, 50%, 75%, serta max dari setiap variabel. Dengan ini, penulis mampu menyimpulkan informasi statistik dengan memperhatikan standar deviasi tertinggi yang terdapat pada variabel "Juni" serta standar deviasi terendah pada variabel "Mei", serta informasi statistik lainnya. Selanjutnya, penulis tertarik untuk mengetahui nomor indeks beserta tipe data yang didapati dalam dataset tersebut.

```

dfbrg.info()

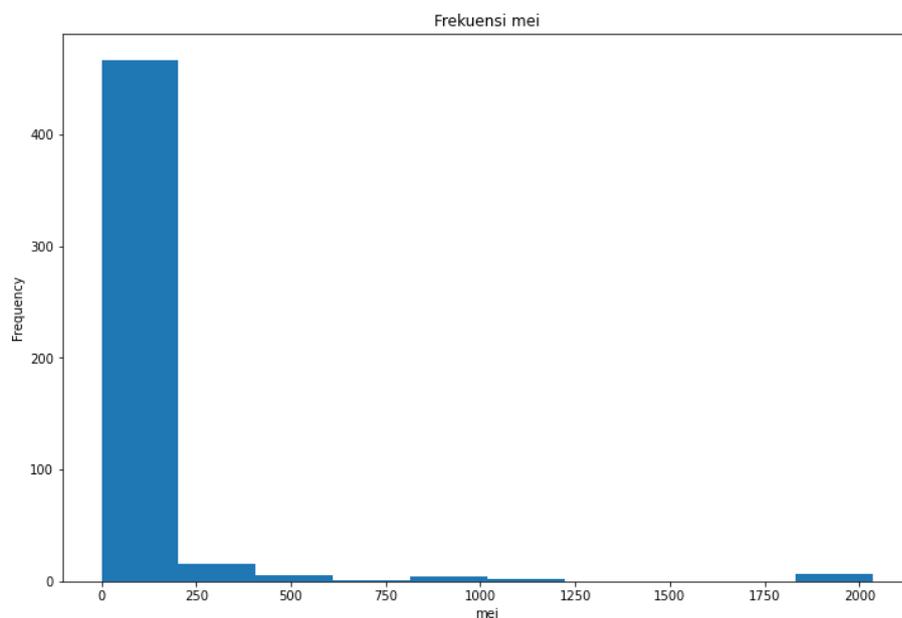
[5] ✓ 0.1s

... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   mei      500 non-null    int64
1   Juni     500 non-null    int64
2   juli     500 non-null    int64
dtypes: int64(3)
memory usage: 11.8 KB

```

**Gambar 4.6 Tampilan info dari Dataset**

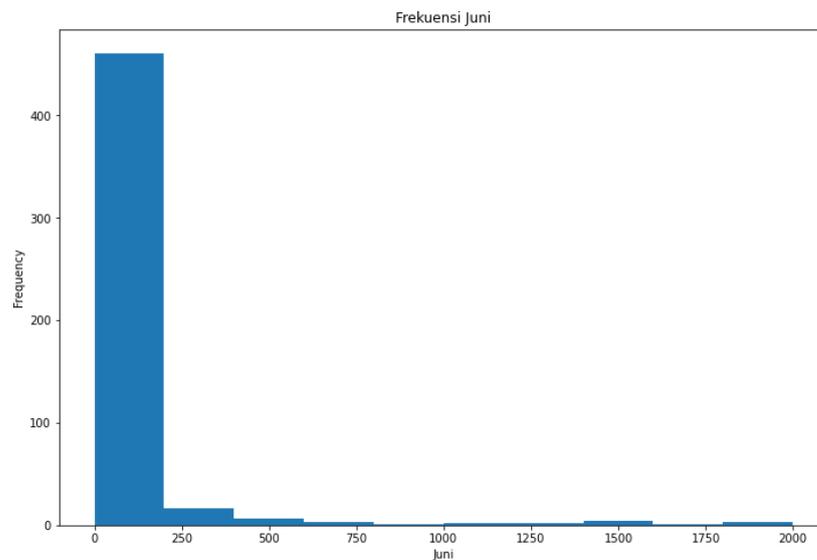
Sesudah informasi tentang setiap variabel beserta tipe datanya muncul, penulis bisa mengeksplorasi kolom-kolom yang berada pada dataframe beserta jumlah data yang terdapat dalam setiap variabel. Setelah langkah tersebut, penulis berkeinginan untuk mengamati frekuensi dari tiga variabel, yaitu "mei", "Juni", serta "juli", dengan cara menghasilkannya dalam bentuk histogram supaya visualisasinya terlihat jelas dan mudah dipahami. Untuk itu, penulis mampu mempergunakan perintah berikut untuk memeriksa frekuensi penjualan dari variabel "mei".



**Gambar 4.7 Tampilan Histogram bulan “mei”**

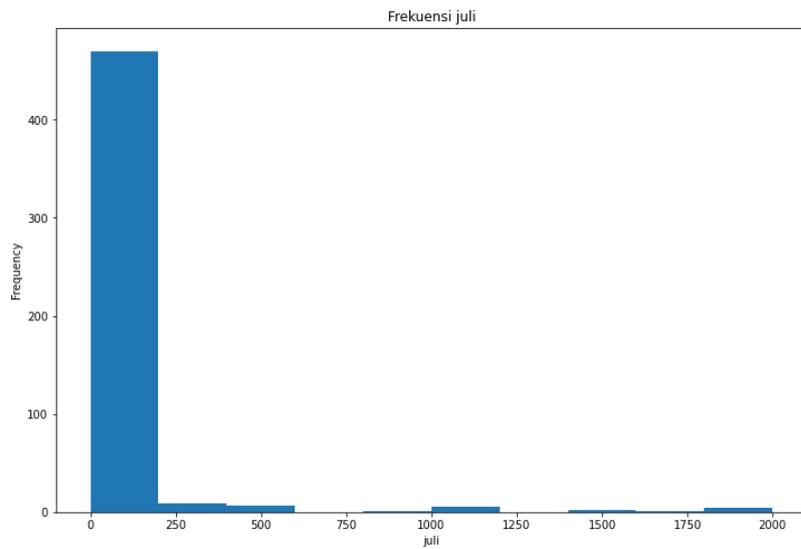
Mampu diamati bahwasanya hasil dari *histogram* pada variabel “mei” frekuensi

penjualan tertinggi berada pada angka 0 hingga 150, maka dari itu penulis dapat menarik kesimpulan bahwa pada bulan mei barang terjual mulai dari 0 hingga 150 barang. Sesudah itu penulis ingin mengamati frekuensi pada variabel berikutnya yakni variabel “juni”. Hasilnya pun sama angka penjualan tertinggi mulai dari 0 hingga 150 barang.



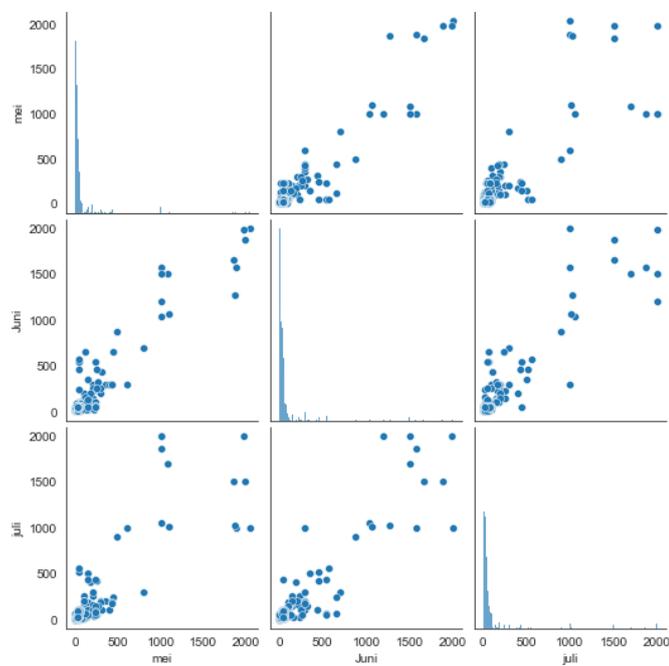
**Gambar 4.8 Tampilan Histogram bulan “juni”**

Tampilan histogram pada variable “juli” frekuensi penjualan barang tidak jauh berbeda dengan bulan mei dan juni yaitu angka penjualan barang mulai dari 0 hingga 150 barang ter:



**Gambar 4.9 Tampilan Histogram bulan “juli”**

Sesudah itu penulis ingin menganalisa keempat variabel tersebut mempergunakan *Multivariate Analysis* dengan *Pairplot* supaya memahami korelasi ataupun hubungan antar 2 buah variabel. Dengan begitu pengguna lebih mudah mengamati korelasi atau hubungan antar variabel.



**Gambar 4.10 Tampilan Pairplot dari Multivariate Analysis**

Pada Gambar 4.8 terlihat hasil dari *Multivariate Analysis* dengan mempergunakan

*Pairplot*, maka sebab itu penulis mampu mengamati secara visualisasi hubungan antar ketigat variabel tersebut. Sesudah itu penulis menampilkan *data frame* yang baru dengan menggunakan perintah

```
dfbrg=dfbrg.drop(["no","nama_barang","satuan"],axis=1)
dfbrg
```

[4] ✓ 0.1s

	mei	Juni	juli
0	0	2	7
1	12	7	3
2	800	700	300
3	10	8	12
4	25	33	34
...	...	...	...
495	28	44	52
496	43	45	43
497	24	35	63
498	23	45	85
499	23	46	25

500 rows × 3 columns

**Gambar 4.11** Tampilan *Data Frame* yang baru

Pada ilustrasi berikut, terlihat sebuah data frame yang sudah dimutakhirkan setelah proses penghapusan data outlier dilakukan pada masing-masing variabel. Perubahan terutama terlihat dari jumlah baris antara data frame sebelumnya dan yang baru, dimana jumlah baris pada data frame baru mencapai 500. Dengan demikian, dapat disimpulkan bahwa tahap pengolahan outlier telah berhasil dilakukan. Langkah selanjutnya adalah mengkonversi data frame ini ke dalam bentuk array guna memperhatikan rentang atau selisih nilai antar variabel dengan menggunakan perintah.

```
array_x = np.array(dfbrg_x)
array_x
```

[10] ✓ 0.1s

```
... array([[ 0,  2,  7],
          [12,  7,  3],
          [800, 700, 300],
          ...,
          [ 24,  35,  63],
          [ 23,  45,  85],
          [ 23,  46,  25]], dtype=int64)
```

### Gambar 4.12 Tampilan Halaman Data Distribusi

Dengan mentransformasikan data frame menjadi format array sebagaimana ditunjukkan pada ilustrasi, hal ini akan menghasilkan kemudahan dalam melihat perbedaan atau jarak nilai antar variabel. Penulis mengaplikasikan proses penskalaan data bertujuan untuk menciptakan rentang nilai antara 0 dan 1 guna memfasilitasi pengklasteran yang lebih efisien dan mudah dilakukan.

```

scaler_fit = MinMaxScaler()
scaled_x = scaler_fit.fit_transform(array_x)
scaled_x

```

[13] ✓ 0.1s

```

... array([[0.          , 0.001       , 0.0035      ],
          [0.00589391, 0.0035      , 0.0015      ],
          [0.39292731, 0.35        , 0.15        ],
          ...,
          [0.01178782, 0.0175      , 0.0315      ],
          [0.01129666, 0.0225      , 0.0425      ],
          [0.01129666, 0.023       , 0.0125      ]])

```

### Gambar 4.13 Tampilan Scalling data

Pada Gambar 21, terlihat bahwasanya rentang nilai antar variabel diubah menjadi skala 0 hingga 1, sehingga mempermudah pelaksanaan proses pengelompokan. Setelah itu, pengguna memasuki tahap Clustering K-means dengan memulainya dari menentukan jumlah cluster menggunakan perintah yang ada seperti yang dijelaskan dalam lampiran 10.

Perintah pada lampiran tersebut berfungsi untuk menetapkan dan mengatur parameter fungsi K-means, misalnya menetapkan jumlah kluster serta keadaan acak. Menetapkan jumlah kluster didasarkan pada data sebelumnya yang merupakan data scaled\_x. Setelah itu, penulis mencari nilai pusat dari setiap kluster dengan mempergunakan instruksi yang diberikan.



```

new_df = pd.DataFrame(data= dfbrg, columns = ['mei','Juni','juli'])
new_df['kluster']=kmeans.labels_
new_df

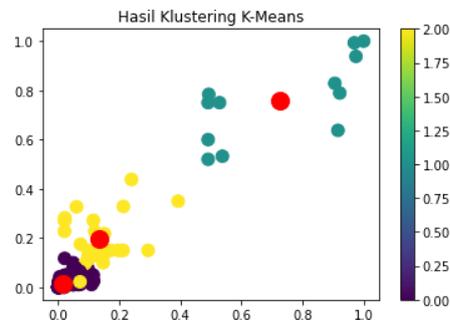
```

	mei	Juni	juli	kluster
0	0	2	7	0
1	12	7	3	0
2	800	700	300	2
3	10	8	12	0
4	25	33	34	0
...	...	...	...	...
495	28	44	52	0
496	43	45	43	0
497	24	35	63	0
498	23	45	85	0
499	23	46	25	0

500 rows × 4 columns

**Gambar 4.16 Source code menambahkan kolom cluster**

Di dalam gambar tersebut, kolom cluster juga sudah diperbarui dengan menggunakan informasi dari label K-means sebelumnya. Untuk memperlihatkan secara visual hasil klaster yang sudah diolah dengan cara yang jelas serta kentara, penulis membuat grafik Scatterplot.



**Gambar 4.17 Grafik Scatter plot hasil akhir**

Dari gambaran scatterplot yang sudah disusun dalam Gambar 4.13, kita dapat dengan mudah melihat tiga kelompok data beserta anggotanya dengan sangat jelas serta secara langsung.

```
new_df.groupby('kluster').count()
```

✓ 0.1s

	mei	Juni	juli
kluster			
0	461	461	461
1	12	12	12
2	27	27	27

**Gambar 4.18 Menampilkan jumlah data dari setiap *cluster***

Dari Gambar 4.18 kita dapat melihat jumlah data dari setiap *cluster*, *cluster* 0 merupakan data penjualan terendah yaitu barang kurang laku jumlah datanya 461 data, *cluster* 1 data penjualan sedang yaitu barang laku jumlah datanya 12 data dan *cluster* 2 data penjualan tertinggi yaitu barang sangat laku jumlah datanya 27 data. Berikut data-data berdasarkan hasil *clusternya*.

**Tabel 4.1 Hasil *Cluster***

<i>Cluster</i> 0	
no	nama_barang
1	Pintu KM/WC
2	Kuas Roll
4	Stop Kran 1 inch Honsu
5	Stop Kran 1/2 inch Honsu
6	Stop Kran 1 inch Sanwell
7	Stop Kran 1 1/2 inch Sanwell
8	Kran Air Cuci Piring Plastik
9	Kran Air Cuci Piring wanly
10	Kran Air Cuci Piring Padova
11	Kran Air Cuci Piring Izumo
12	Floor Drain Stainless
13	Floor Drain Plastik
14	Karpet Talang Air Tebal 110 cm
15	Karpet Talang Air Tebal 90 cm
16	Karpet Talang Air Tebal 50 cm
17	Terpal Plastik uk 3x4 meter
18	Terpal Plastik uk 4X6 meter
19	Terpal Plastik uk 5X7 meter
20	Kran Air plastik Baling
***	***
500	Buis Beton 50 cm ( 1 M' )

**Tabel 4.1 Hasil Cluster (Lanjutan)**

<i>cluster 1</i>	
no	nama_barang
112	Semen Baturaja
260	Batu Pinggir Beton 15 x 35 x 50 ( K-225 )
262	Batako Kecil 8 x 10 x 20
263	Batako Besar 8 x 20 x 30
264	Con Blok 8 x 20 x 40
265	Bata Merah Bakar kelas I
266	Bata Merah Bakar kelas I I
267	Bata Merah Oven ( Klingker )
268	Roster Beton 20 x 20
269	Roster Beton 30 x 30
270	Grass Blok 20 x 20
277	Genteng

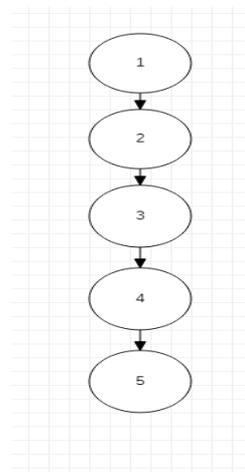
<i>cluster 2</i>	
no	nama_barang
3	Amplas
28	Ember Cor Kecil
62	Sendok Semen Bulat
85	Sarung Tangan Tukang Karet
87	Benang Kasur
177	Tutup Pipa Saringan 4 inch
189	Knie Grest 1/2 inch
225	Besi Cor
227	Benang Nilon
254	Batu Pecah Mesin 5/ 7
272	Paving Blok Natural 8 cm
273	Paving Blok Warna 8 cm
274	Paving Blok Natural 6 cm
275	Paving Blok Warna 6 cm
276	Canstin Paving Blok
280	Readymix Beton K500, Tanpa Pompa ( Selang )
289	Tanah liat
388	Plafond hardpleks 5 mm 120 x 240
452	Keramik 20 x 25 Dinding KM KW I DN Corak Roman
453	Keramik 20 x 25 KW I DN Putih Polos Roman
465	Plin Taraso 10 x 30
468	Granit Alam DN Ukuran Besar
472	Granito Tile Essenza Ukuran 40 x 40 UnPolis
482	Bata Karawang
484	Plint keramik 10 x 20
485	Plint keramik 10 x 25

### 4.1.2 Pengujian

Pengujian sistem yang telah direncanakan bertujuan untuk mengevaluasi dengan cermat sejauh mana sistem tersebut mematuhi perancangan awalnya. Ini dilangsungkan dengan penekanan khusus pada kode sumber program. Saat melakukan pengujian ini dengan White box testing, fokus utamanya yakni pada alur logika yang terdapat dalam kode program yang telah dibuat.

Metode yang dipergunakan pada menguji program ini adalah white box testing, yang merupakan pendekatan yang melibatkan penguji untuk memahami dan menguji struktur internal program. Setelah melakukan uji coba menyeluruh, hasilnya menunjukkan bahwa program dapat berjalan sesuai dengan yang diinginkan. Penulis juga menggunakan pengujian white box sebagai metode pengujian untuk klasifikasi jenis barang pada sebuah toko bangunan bernama Galunggung dalam topik tugas akhirnya.

Pengujian white box testing dilangsungkan dengan fokus pada struktur internal program, di mana setiap baris kode ataupun perintah dianalisis secara terperinci untuk mendeteksi kesalahan yang mungkin terjadi saat program dijalankan. Hasilnya menunjukkan bahwa white box testing memanfaatkan petunjuk untuk menciptakan program yang diinginkan dan efisien. Sehingga, penulis memutuskan untuk menggunakan Analisis Grafik Alur (Flowgraph) dalam proses analisis. Berikut ini adalah contoh Gambar Analisis Grafik Alur.



### Gambar 4.19 Flowgraph

Gambar 4.19 memperlihatkan sebuah Flowgraph yang dipergunakan oleh penulis dalam pengujian program Clustering K-means. Flowgraph tersebut dipergunakan untuk mengukur tingkat kompleksitas logika dari suatu kode program dan dengan menggunakan nilai  $V(G)$ , penguji melangsungkan pengujian jalur basis. Angka  $V(G)$  mewakili jumlah maksimum kasus uji yang perlu dirancang dengan mengidentifikasi kumpulan jalur eksekusi dasar untuk memastikan semua pernyataan dieksekusi setidaknya sekali.

#### 1. Listing Program Clustering K-means Analisis

**Tabel 4.2 Listing Program Clustering K-means**

Node	Source code	Keterangan
1	<pre>import pandas as pd import numpy as np import seaborn as sns import matplotlib.pyplot as plt from sklearn.cluster import KMeans from sklearn.preprocessing import MinMaxScaler</pre>	Dengan mengimpor beberapa library, pengguna mampu membuat penulisan kode atau perintah yang ingin mereka gunakan lebih sederhana dan ringkas.
2	<pre>#Mengimport File Dataset df = pd.read_csv("dataset.csv") df.head()</pre>	Memasukkan dataset ke dalam notebook Jupyter untuk melakukan proses Clustering K-means dengan tujuan melakukan Clustering K-means.
3.	<pre>#Describing Data df.describe()  #Data Info df.info()  #Cleaning Data df.fillna(method='ffill')  #Mengambil Variabel Tertentu df= df[['Mei','Juni','Juli']]  #Univariate Analysis pada Variabel Mei</pre>	Pada tahap ini, dilangsungkan analisis serta pemilihan dataset sebelum proses clustering K-means dilakukan. Langkah-langkah di atas mencakup penggunaan kode sumber untuk meninjau informasi dataset, memvisualisasikan variabel dalam bentuk histogram, dan proses lainnya.

**Tabel 4.3 Listing Program Clustering K-means (Lanjutan)**

<i>Node</i>	<i>Source code</i>	<i>Keterangan</i>
	<pre>plt.figure(figsize=(12,8)) plt.hist(df['mei']) plt.title('Frekuensi mei') plt.ylabel('Frequency') plt.xlabel('mei') plt.show()  #Univariate Analysis pada Variabel Juni plt.figure(figsize=(12,8)) plt.hist(df['Juni']) plt.title('Frekuensi Juni') plt.ylabel('Frequency') plt.xlabel('Juni') plt.show()  #Univariate Analysis pada Variabel Juli plt.figure(figsize=(12,8)) plt.hist(df['juli']) plt.title('Frekuensi Juli') plt.ylabel('Frequency') plt.xlabel('juli') plt.show()  #Multivariate Analysis sns.set_style("white") sns.pairplot(df, vars=['mei', 'Juni', 'juli'], diag_kind='hist') #Remove Data Outlier  #Mengubah Data menjadi bentuk Array df = np.array(df) print(df)  #Scalling Data scaler = MinMaxScaler() x_scaled = scaler.fit_transform(df) x_scaled</pre>	

**Tabel 4.2 Listing Program Clustering K-means (Lanjutan)**

<i>Node</i>	<i>Source code</i>	<i>Keterangan</i>
4	<pre>#Menentukan Jumlah Cluster dan Random State kmeans = Kmeans (n_clusters = 3, random_state=123)  #Mencetak atau menampilkan nilai pusat dari masing – masing cluster print (kmeans.cluster_centers_)  #Mencetak atau menampilkan cluster atau labels dari masing – masing data print (kmeans.labels_)</pre>	<p>Pada tahap ini, source code memiliki peran penting dalam menentukan sejumlah Cluster serta Random State yang diperlukan. Melalui source code ini pula, dilakukan pengaturan nilai tengah dari setiap cluster untuk kemudian menampilkan cluster atau label yang sesuai dengan setiap data yang ada.</p>
5	<pre>#Menambahkan Kolom Cluster new_df = pd.DataFrame(data=df2, columns=['mei', 'juni', 'juli']) new_df['kluster'] = kmeans.labels_ new_df  #Membuat Grafik Scatterplot output = plt.scatter(x_scaled[:,0], x_scaled[:,1], s = 100, c = new_df.kluster, marker = "o", alpha = 1, ) centers = kmeans.cluster_centers_ plt.scatter(centers[:,0], centers[:,1], c='red', s=200, alpha=1 , marker="s"); plt.title("Hasil Klustering K-means") plt.colorbar (output) plt.show()</pre>	<p>Pada fase ini, source code yakni hasil dari proses Clustering K-means. Setelah berhasil mengelompokkan data, hasil klaster tersebut dimasukkan ke dalam kolom klaster pada kerangka data. Selanjutnya, data diproses untuk dibentuk menjadi scatterplot sehingga hasil Clustering K-means dapat dengan jelas terlihat dan dipahami secara visual.</p>

## 2. Kompleksitas Siklomatik

Dalam memutuskan pengukuran kompleksitas logika pada suatu program *Clustering K-means* maka didapati dengan perhitungan sebagaimana berikut :

$$V(G) = E - N + 2$$

Keterangan :

E = Jumlah *edge* pada *flowgraph*

N = Jumlah *node* pada *flowgraph*

R = Jumlah *region* pada *flowgraph*

Sehingga kompleksitas siklomatis yang diperoleh adalah :

a. *Flowgraph* mempunyai :

*Region* (R)    2

*Node* (N)     5

*Edge* (E)     4

b.  $V(G) = E - N + 2$

$$= 4 - 5 + 2$$

$$= 1$$

Maka jumlah *cyclopatic complexity* pada *flowgraph* ini yakni 1.

### 3. Pengujian *Basis Path*

**Tabel 4.4 Pengujian *Basis Path***

Jalur	Input	Proses	Keterangan
Jalur 1	<i>Dataset</i> berhasil diolah ke program <i>Clustering K-means</i>	1-2-3-4-5	<i>Dataset</i> berhasil diolah ke program <i>Clustering K-means</i> dengan <i>system</i> sesuai rule yang terbentuk.