

## **BAB III**

### **METODOLOGI PENELITIAN**

Pada bab ini akan membahas tentang metodologi penelitian yang dilakukan dalam Tugas Akhir ini.

#### **3.1 Subjek dan Objek Penelitian**

Berdasarkan latar belakang yang telah diuraikan sebelumnya, dalam penelitian ini dilakukan penelitian tentang analisis data pengiriman barang ekspedisi untuk mengelompokkan potensi barang pengiriman menggunakan metode *K-Means Clustering* studi kasus TIKI KC Purbalingga. Adapun subjek dari penelitian ini adalah TIKI KC Purbalingga, sedangkan objek dari penelitian ini adalah data pengiriman barang di TIKI KC Purbalingga pada periode 2018 - 2021.

#### **3.2 Alat dan Bahan Penelitian**

Penelitian ini membutuhkan beberapa spesifikasi minimum dari perangkat keras dan perangkat lunak.

a. Spesifikasi kebutuhan perangkat keras

Perangkat keras yang diperlukan dalam penelitian ini yaitu Laptop / Komputer untuk mengakses data pengiriman barang TIKI KC Purbalingga dengan spesifikasi minimum yang dibutuhkan yaitu memiliki *Processor* Intel Core i3 dan 2GB RAM.

b. Spesifikasi kebutuhan perangkat lunak

Berikut merupakan perangkat lunak yang dibutuhkan untuk melakukan penelitian ini.

1. *Google Collaboratory (Google Colab)* untuk melakukan proses *clustering K-Means* data penelitian.
2. *Microsoft Excel* untuk mengemas dan menampilkan data pengiriman barang pada TIKI kantor cabang Purbalingga sebagai objek penelitian..
3. *Google Drive* untuk mengunggah file excel yang telah diolah dan siap untuk diproses di *Google Collaboratory*.
4. Aplikasi *Microsoft Word* untuk menulis laporan.

### 3.3 Proses Penelitian

Pada penelitian ini digunakan metode *K-Means Clustering*, yang terbagi kedalam beberapa tahap sebagai berikut : (a) Tahap studi literatur, (b) Tahap pengumpulan data, (c) Tahap pengolahan data, (d) Tahap *clustering*, (e) Tahap analisa [12]. Dengan rincian pada Gambar 3.1. berikut ini :



**Gambar 3.1. Gambar Alur Penelitian**

### 3.4 Data

Data yang akan digunakan untuk mengelompokkan adalah data pengiriman barang yang diperoleh dari TIKI KC Purbalingga selama periode 2018 – 2021 sebanyak 34991 data.

### 3.5 Jenis data

Jenis data yang diambil adalah dari file rekap data pengiriman barang TIKI KC Purbalingga dari tahun 2018 – 2021 yang berupa file .xlsx yang kemudian diolah dan diubah kedalam bentuk dokumen yang bereksistensi .csv.

### 3.6 Analisa Data

Secara umum, penelitian ini untuk mengelompokkan potensi barang pengiriman, Data yang dianalisis ini merupakan hasil rekap pengiriman barang oleh *customer* TIKI KC Purbalingga. Data yang digunakan merupakan data jenis barang, total pengiriman, total pendapatan, dan tahun pengiriman, sehingga nantinya pihak terkait dapat mengambil tindakan untuk memetakan barang pengiriman untuk dilakukan peningkatan pelayanan dalam hal pengemasan, penyimpanan, dan pengiriman barang. Berikut ini adalah contoh data yang akan digunakan untuk perhitungan dengan *K-Means Clustering* atau pada Tabel 3.1 :

**Tabel 3.1. Data yang digunakan untuk perhitungan**

Jenis Barang	Total Pengiriman	Total Berat	Total Pendapatan	Tahun
1	1858	4474	105546425	2018
2	2345	5972	143452637	2018
3	1859	4656	109667274	2018
4	339	854	19725326	2018
5	2146	6125	160957719.6	2018
6	1501	4405,5	1160957720	2018
7	1029	2724	70724438	2018
8	265	563	14450775	2018
9	378	989	25570875	2018
10	189	440	10148113	2018

Jenis Barang	Total Pengiriman	Total Berat	Total Pendapatan	Tahun
11	0	0	0	2018
1	1598	4268	133819780	2019
2	1998	4237	101150792	2019
3	1591	3619	92953870	2019
4	282	632	15987904	2019
5	1850	4306,5	122894149	2019
6	1290	3391	108298581	2019
7	879	2287	67166991	2019
8	235	531	15563725	2019
9	324	728	20009000	2019
10	160	416	13700450	2019
11	0	0	0	2019
1	1000	2461	79075750	2020
2	1258	3022	91569284	2020
3	1001	2644	91058525	2020
4	181	382	8745680	2020
5	1141	2757	82794374	2020
6	801	1901	59044284	2020
7	555	1408	47212720	2020
8	143	316	9404650	2020
9	201	542	18087250	2020
10	101	199	5228450	2020
11	0	0	0	2020
1	979	1967	41984950	2021
2	1174	2757	76169164.85	2021
3	1054	2383	67457700	2021
4	150	218	3419750	2021
5	1000	5423	241974183.6	2021
6	805	3488	133682700	2021
7	503	1007	22536850	2021
8	198	343	4467350	2021

Jenis Barang	Total Pengiriman	Total Berat	Total Pendapatan	Tahun
9	251	707	24473400	2021
10	85	126	2022650	2021
11	4	120	4023050	2021

### 3.6.1 Transformasi Data

Agar data dapat diolah dengan menggunakan metode *K-Means clustering*, maka data yang berjenis data nominal seperti jenis barang kedalam bentuk angka yang dapat dilihat pada Tabel 3.2 :

**Tabel 3.2 Inisiasi Jenis Barang**

Jenis Barang	Inisiasi
Herbal	1
Baju	2
Bulu mata	3
Surat berharga	4
Knalpot	5
ACC motor	6
Makanan	7
Aksesoris	8
Kosmetik	9
Sepatu	10
Kendaraan	11

Keseluruhan data, seperti total pengiriman, total berat, total pendapatan dijumlahkan dan dikelompokkan berdasarkan tahun pengiriman dengan menggunakan rumus excel :

Total Pengiriman = SUMIF(Qbarang:Qbarang;"Barang";Gpcs:Gpcs)

Total Berat = SUMIF(Qbarang:Qbarang;"Barang";Hwgh:Hwgh)

Total Pendapatan = SUMIF(Qbarang:Qbarang;"Barang";Psetor:Psetor)

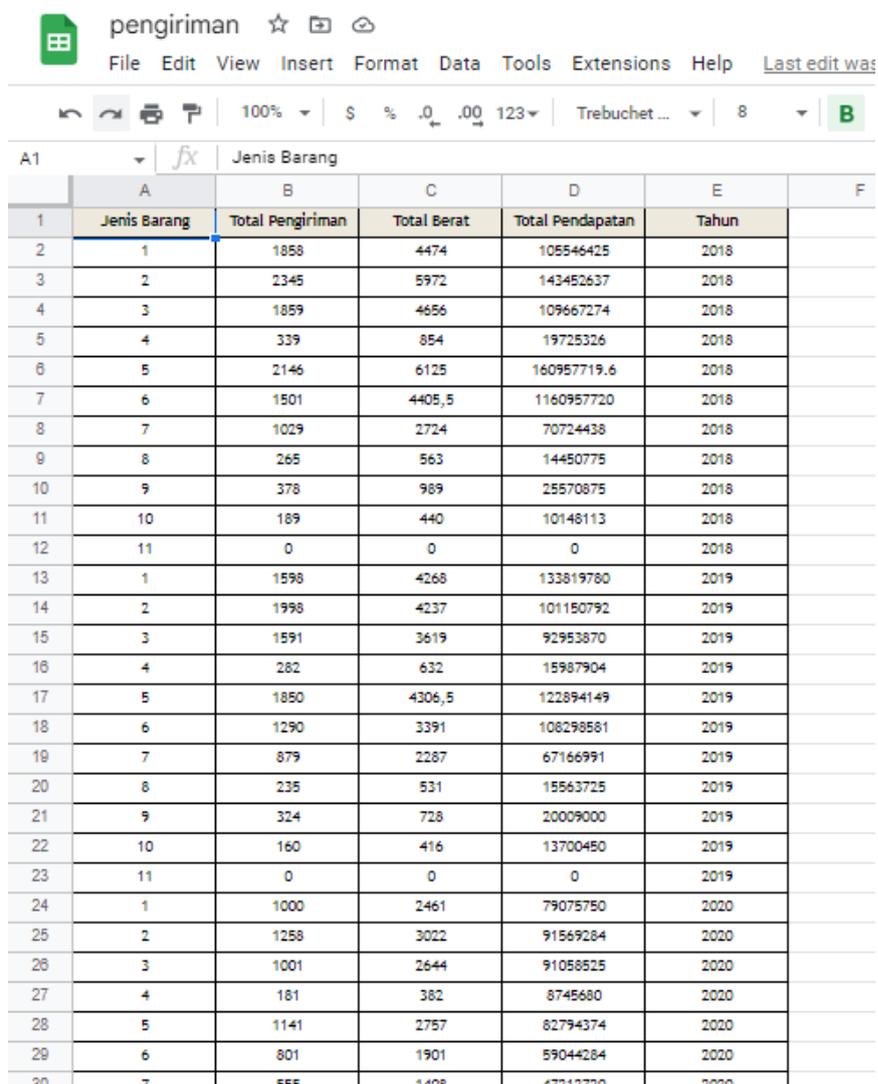
Tabel 3.2. Potongan data yang didapatkan dari TIKI KC Purbalingga

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Trx Date	Time	No Resi	Origin	Dest	Service	Pcs	C Wgh	Shipment	Surcharge	Ins	Pack	Disc	Disc Value	Total	Setor	Barang	Shipment History
2	02-01-2018	09:26:49	30221921202	PWT03.00	CGK01.00	REG	1,	4,	50.000,00	0,00	0,00	0,00	20, %	10.000,	50.000,00	40.000,	Herbal	Ter kirim
3	02-01-2018	09:32:08	30221921203	PWT03.00	BDJ04.00	REG	1,	2,	65.000,00	0,00	0,00	0,00	15, %	9.750,	65.000,00	55.250,	Baju	Ter kirim
4	02-01-2018	09:47:36	30221921204	PWT03.00	BGR01.00	ONS	1,	1,	17.000,00	0,00	0,00	0,00	20, %	3.400,	17.000,00	13.600,	bulu mata	Ter kirim
5	02-01-2018	09:48:55	30221921205	PWT03.00	BKI01.00	ONS	1,	1,	17.000,00	0,00	0,00	0,00	20, %	3.400,	17.000,00	13.600,	bulu mata	Ter kirim
6	02-01-2018	10:11:51	30221921206	PWT03.00	GRT70.00	REG	1,	2,	29.500,00	12.500,00	0,00	0,00	15, %	6.300,	42.000,00	35.700,	Baju	Ter kirim
7	02-01-2018	10:13:29	30221921207	PWT03.00	BKI75.00	REG	1,	1,	12.000,00	0,00	0,00	0,00	15, %	1.800,	12.000,00	10.200,	Surat Berharga	Ter kirim
8	02-01-2018	10:15:20	30221921208	PWT03.00	AMI01.00	REG	1,	1,	28.000,00	0,00	0,00	0,00	15, %	4.200,	28.000,00	23.800,	Herbal	Ter kirim
9	02-01-2018	11:16:08	30221921209	PWT03.00	JMS02.67	REG	1,	3,	126.000,00	29.300,00	0,00	8.000,00	15, %	23.295,	163.300,00	132.005,	Knalpot	Ter kirim
10	02-01-2018	11:25:19	30221921210	PWT03.00	CGK01.00	ONS	1,	2,	28.000,00	0,00	0,00	8.000,00	20, %	5.600,	36.000,00	22.400,	bulu mata	Ter kirim
11	02-01-2018	11:26:25	30221921211	PWT03.00	BDQ72.02	ONS	1,	3,	37.000,00	9.000,00	0,00	10.000,00	20, %	9.200,	56.000,00	36.800,	Baju	Ter kirim
12	02-01-2018	11:28:21	30221921212	PWT03.00	TGR76.00	ONS	1,	3,	45.000,00	0,00	0,00	10.000,00	20, %	9.000,	55.000,00	36.000,	Baju	Ter kirim
13	02-01-2018	11:35:43	30221921213	PWT03.00	BGR01.00	REG	1,	1,	12.000,00	0,00	0,00	0,00	15, %	1.800,	12.000,00	10.200,	Herbal	Ter kirim
14	02-01-2018	12:02:15	30221921214	PWT03.00	CGK01.00	ONS	1,	1,	17.000,00	0,00	12.250,00	0,00	20, %	3.400,	29.250,00	25.850,	Baju	Ter kirim
15	02-01-2018	12:40:38	30221921215	PWT03.00	BDO01.00	ONS	1,	1,	17.000,00	0,00	0,00	0,00	20, %	3.400,	17.000,00	13.600,	Baju	Ter kirim
16	02-01-2018	12:43:20	30221921216	PWT03.00	PWT02.00	REG	1,	2,	34.000,00	0,00	0,00	8.000,00	15, %	5.100,	42.000,00	28.900,	Makanan	Ter kirim
17	02-01-2018	12:46:57	30221921217	PWT03.00	TKG01.00	ONS	1,	2,	54.000,00	0,00	0,00	0,00	20, %	10.800,	54.000,00	43.200,	ACC	Ter kirim
18	02-01-2018	12:48:52	30221921218	PWT03.00	CBND1.00	ECO	1,	1,	11.000,00	0,00	0,00	0,00	15, %	1.650,	11.000,00	9.350,	Baju	Ter kirim
19	02-01-2018	12:50:17	30221921219	PWT03.00	AMI01.00	ECO	1,	1,	28.000,00	0,00	0,00	0,00	15, %	4.200,	28.000,00	23.800,	Baju	Ter kirim
20	02-01-2018	12:53:30	30221921220	PWT03.00	SEG01.00	ONS	1,	1,	18.000,00	0,00	0,00	0,00	20, %	3.600,	18.000,00	14.400,	Herbal	Ter kirim
21	02-01-2018	12:55:02	30221921221	PWT03.00	TKG13.05	REG	1,	2,	54.000,00	0,00	0,00	0,00	15, %	8.100,	54.000,00	45.900,	Herbal	Ter kirim
22	02-01-2018	13:26:20	30221921222	PWT03.00	TN101.00	REG	1,	7,	203.000,00	0,00	0,00	0,00	15, %	30.450,	203.000,00	172.550,	Knalpot	Ter kirim

### 3.6.2 Pengolahan Data

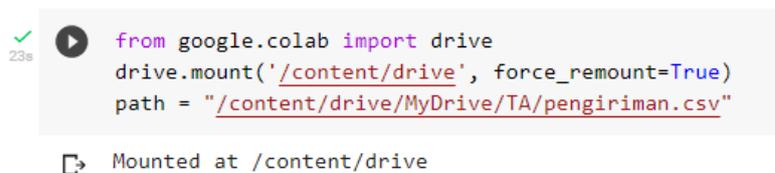
Setelah semua data barang tahun 2018 sampai dengan 2021 ditransformasi dalam bentuk angka dan keseluruhan data telah dijumlahkan dan dikelompokkan per tahun, maka selanjutnya data – data tersebut sudah bisa dikelompokkan dengan menggunakan algoritma *K-Means Clustering* menggunakan *Google Collaboratory*. Namun sebelum itu, tabel data yang telah dikelompokkan tadi haruslah di *upload* ke *Google Drive* agar dapat diolah di *Google Collaboratory* nantinya, data yang di *upload* digambarkan pada Tabel 3.4.

**Tabel 3.3.** Data yang telah diunggah ke *Google Drive*



	A	B	C	D	E	F
1	Jenis Barang	Total Pengiriman	Total Berat	Total Pendapatan	Tahun	
2	1	1858	4474	105546425	2018	
3	2	2345	5972	143452637	2018	
4	3	1859	4656	109667274	2018	
5	4	339	854	19725326	2018	
6	5	2146	6125	160957719.6	2018	
7	6	1501	4405,5	1160957720	2018	
8	7	1029	2724	70724438	2018	
9	8	265	563	14450775	2018	
10	9	378	989	25570875	2018	
11	10	189	440	10148113	2018	
12	11	0	0	0	2018	
13	1	1598	4268	133819780	2019	
14	2	1998	4237	101150792	2019	
15	3	1591	3619	92953870	2019	
16	4	282	632	15987904	2019	
17	5	1850	4306,5	122894149	2019	
18	6	1290	3391	108298581	2019	
19	7	879	2287	67166991	2019	
20	8	235	531	15563725	2019	
21	9	324	728	20009000	2019	
22	10	160	416	13700450	2019	
23	11	0	0	0	2019	
24	1	1000	2461	79075750	2020	
25	2	1258	3022	91569284	2020	
26	3	1001	2644	91058525	2020	
27	4	181	382	8745680	2020	
28	5	1141	2757	82794374	2020	
29	6	801	1901	59044284	2020	
30	7	555	1408	47217720	2020	

Setelah berhasil diunggah ke *Google Drive*, selanjutnya masuk ke tahap mengkoneksikan *Google Drive* dan juga identifikasi file yang akan dianalisis dengan *Google Collaboratory* yang digambarkan pada Gambar 3.5.



```

✓ 23s ▶ from google.colab import drive
drive.mount('/content/drive', force_remount=True)
path = "/content/drive/MyDrive/TA/pengiriman.csv"

Mounted at /content/drive

```

**Gambar 3.4.** Import *Drive* dengan *Google Collaboratory*

Selanjutnya, untuk dapat mengolahnya di *Google Collaboratory*, sebelumnya haruslah mengimport library *numpy*, *pandas*, dan *matplotlib*.

```
[6] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from mpl_toolkits import mplot3d
from sklearn.cluster import KMeans
from sklearn import metrics
from sklearn.preprocessing import RobustScaler
from scipy.spatial.distance import cdist
from yellowbrick.cluster import SilhouetteVisualizer
```

**Gambar 3.5. Import library yang akan digunakan**

Setelah itu, tabel yang akan dianalisis dicoba untuk ditampilkan dengan perintah pada Gambar 3.7, dimana *df* merupakan inisiasi dari data. Saat setelah dijalankan (di *run*), maka akan tampil tabel yang dipanggil tadi, yaitu pada Gambar 3.8 yang dimana tabel tersebut merupakan hasil olahan data pengiriman barang dan telah di upload di *.csv*.

```
df = pd.read_csv(path)
df
```

**Gambar 3.6. Membaca tabel**

	Jenis Barang	Total Pengiriman	Total Berat	Total Pendapatan	Tahun
0	1	1858	4474	1.055464e+08	2018
1	2	2345	5972	1.434526e+08	2018
2	3	1859	4656	1.096673e+08	2018
3	4	339	854	1.972533e+07	2018
4	5	2146	6125	1.609577e+08	2018
5	6	1501	4405.5	1.160958e+09	2018
6	7	1029	2724	7.072444e+07	2018
7	8	265	563	1.445078e+07	2018
8	9	378	989	2.557088e+07	2018
9	10	189	440	1.014811e+07	2018
10	11	0	0	0.000000e+00	2018
11	1	1598	4268	1.338198e+08	2019
12	2	1998	4237	1.011508e+08	2019
13	3	1591	3619	9.295387e+07	2019
14	4	282	632	1.598790e+07	2019
15	5	1850	4306.5	1.228941e+08	2019
16	6	1290	3391	1.082986e+08	2019
17	7	879	2287	6.716699e+07	2019

**Gambar 3.7. Menampilkan tabel**

Sekarang memasuki tahap mencoba untuk menampilkan grafik data, yaitu dengan mengambil tiga parameter bagian dari dataset dimana pada kali ini penulis menggunakan jenis barang (x1), total pengiriman (x2), dan total pendapatan (x3). Kemudian memindahkan tiap kolom ke variabel grafik dengan perintah `x1 = np.array(df["Jenis Barang"])`, dst. Yang kemudian mempersiapkan grafik datanya menggunakan perintah `ax = plt.axes(projection = '3d')`, lalu mendefinisikannya menjadi label x, y, z dengan perintah `ax.set_xlabel("Jenis Barang")`, dst sampai dengan `zlabel`, setelah itu gunakan perintah `ax.scatter(x1,x2,x3)`. Yang terakhir menampilkan grafiknya dengan menggunakan `plt.title('Data Pengiriman')` yaitu nama grafiknya, dan `plt.show()` untuk menampilkannya.

```
✓ [8] #Ambil bagian data dari dataset
05 x = np.array(df[["Jenis Barang", "Total Pengiriman", "Total Pendapatan"]])
print("Bagian data : ")
print(x)

#Membuat grafik data
#1. Memindah tiap kolom data ke variabel grafik
x1 = np.array(df["Jenis Barang"])
x2 = np.array(df["Total Pengiriman"])
x3 = np.array(df["Total Pendapatan"])

#2. Menyiapkan grafik data
ax = plt.axes(projection = '3d')
#ax adalah obyek 3d yang akan diletakkan di grafik data

#3. Mendefining axes
ax.set_xlabel("Jenis Barang")
ax.set_ylabel("Total Pengiriman")
ax.set_zlabel("Total Pendapatan")
ax.scatter(x1, x2, x3)
#data x, y, dan z ditempelkan di obyek ax

#4. Menampilkan grafik
plt.title('Data pengiriman')
plt.show()
```

**Gambar 3.8. Menampilkan grafik**

```

0a  ▶ Bagian data :
    □ [[1.00000000e+00 1.85800000e+03 1.05546425e+08]
      [2.00000000e+00 2.34500000e+03 1.43452637e+08]
      [3.00000000e+00 1.85900000e+03 1.09667274e+08]
      [4.00000000e+00 3.39000000e+02 1.97253260e+07]
      [5.00000000e+00 2.14600000e+03 1.60957720e+08]
      [6.00000000e+00 1.50100000e+03 1.16095772e+09]
      [7.00000000e+00 1.02900000e+03 7.07244380e+07]
      [8.00000000e+00 2.65000000e+02 1.44507750e+07]
      [9.00000000e+00 3.78000000e+02 2.55708750e+07]
      [1.00000000e+01 1.89000000e+02 1.01481130e+07]
      [1.10000000e+01 0.00000000e+00 0.00000000e+00]
      [1.00000000e+00 1.59800000e+03 1.33819780e+08]
      [2.00000000e+00 1.99800000e+03 1.01150792e+08]
      [3.00000000e+00 1.59100000e+03 9.29538700e+07]
      [4.00000000e+00 2.82000000e+02 1.59879040e+07]
      [5.00000000e+00 1.85000000e+03 1.22894149e+08]
      [6.00000000e+00 1.29000000e+03 1.08298581e+08]
      [7.00000000e+00 8.79000000e+02 6.71669910e+07]
      [8.00000000e+00 2.35000000e+02 1.55637250e+07]
      [9.00000000e+00 3.24000000e+02 2.00090000e+07]
      [1.00000000e+01 1.60000000e+02 1.37004500e+07]
      [1.10000000e+01 0.00000000e+00 0.00000000e+00]
      [1.00000000e+00 1.00000000e+03 7.90757500e+07]
      [2.00000000e+00 1.25800000e+03 9.15692840e+07]
      [3.00000000e+00 1.00100000e+03 9.10585250e+07]
      [4.00000000e+00 1.81000000e+02 8.74568000e+06]
      [5.00000000e+00 1.14100000e+03 8.27943740e+07]
      [6.00000000e+00 8.01000000e+02 5.90442840e+07]
      [7.00000000e+00 5.55000000e+02 4.72127200e+07]
      [8.00000000e+00 1.43000000e+02 9.40465000e+06]
      [9.00000000e+00 2.01000000e+02 1.80872500e+07]
      [1.00000000e+01 1.01000000e+02 5.22845000e+06]
      [1.10000000e+01 0.00000000e+00 0.00000000e+00]
      [1.00000000e+00 9.79000000e+02 4.19849500e+07]
      [2.00000000e+00 1.17400000e+03 7.61691648e+07]
      [3.00000000e+00 1.05400000e+03 6.74577000e+07]
      [4.00000000e+00 1.50000000e+02 3.41975000e+06]
      [5.00000000e+00 1.00000000e+03 2.41974184e+08]
      [6.00000000e+00 8.05000000e+02 1.33682700e+08]

```

**Gambar 3.9. Tampilan datanya**

Selanjutnya kita menentukan jumlah  $k$  terbaik dan menggambar elbow dengan perintah yang telah tertera dalam **Gambar 3.11**, kemudian menampilkannya dengan perintah `plt.show()`, sehingga tampilannya akan seperti

Proses perhitungan, yaitu dengan melakukan perhitungan *K-Means* dengan library yang telah diimport sebelumnya. Perhitungan *K-Means* dilakukan dengan Euclidian, lalu menampilkan hasilnya ke dalam sebuah grafik **Gambar 3.11**. Bisa dilihat bahwa pada grafik tersebut titik elbownya 1, 3, 7, maka dapat diperkirakan bahwa jumlah *cluster* optimal ada pada titik – titik tersebut.

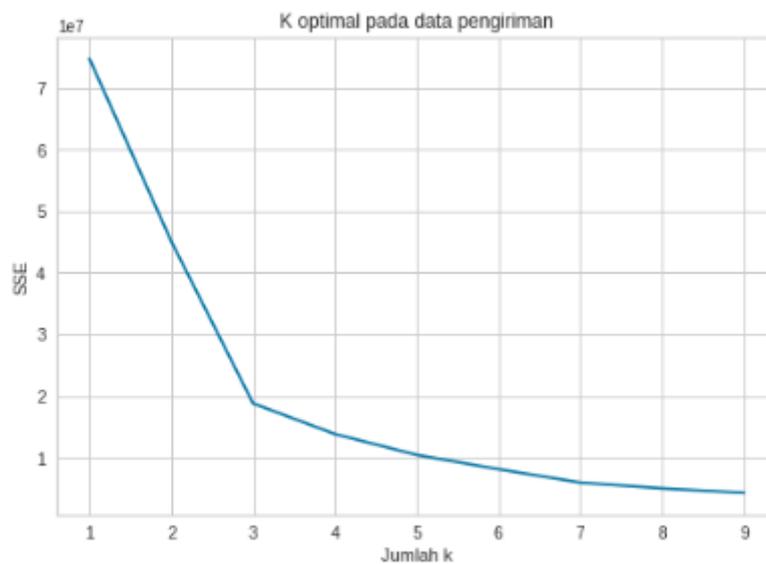
```

[9] #Menentukan jumlah k terbaik
distortions = []
K = range(1, 10)
for k in K:
    kmeanModel = KMeans(n_clusters=k).fit(X)
    kmeanModel.fit(X)
    distortions.append(sum(np.min(cdist(X, kmeanModel.cluster_centers_, 'euclidean'), axis=1)) / X.shape[0])

#Menggambar elbow
plt.plot(K, distortions, 'bx-')
plt.xlabel('Jumlah k')
plt.ylabel('SSE')
plt.title('K optimal pada data pengiriman')
plt.show()

```

**Gambar 3.10.** Menentukan jumlah k terbaik dan menggambar *elbow*



**Gambar 3.11.** Grafik elbow K optimal pada data pengiriman

Dibawah ini atau pada **Gambar 3.13** merupakan tahapan selanjutnya, yaitu *Preprocessing Data* dengan melakukan tahapan *preprocessing data* sebelum melakukan pengolahan atau *clustering data*. Tahapan *preprocessing data* yang dilakukan dengan menggunakan normalisasi dengan *library* RobustScaler.

```

[13] #Preprocessing: Normalisasi data dengan RobustScaler
scaler = RobustScaler()
x_scaled = scaler.fit_transform(X)
x_scaled

```

**Gambar 3.12.** *Preprocessing Data*

```

✓ [10] array([[ -0.83333333,  1.18088566,  0.74153845],
             [ -0.66666667,  1.66825119,  1.20273657],
             [ -0.5       ,  1.18188641,  0.79167608],
             [ -0.33333333, -0.33925444, -0.30263154],
             [ -0.16666667,  1.46910183,  1.41571778],
             [  0.       ,  0.82361771, 13.58253393 ],
             [  0.16666667,  0.35126345,  0.31786555],
             [  0.33333333, -0.41330998, -0.36680607],
             [  0.5       , -0.30022517, -0.23150979],
             [  0.66666667, -0.48936703, -0.41915579],
             [  0.83333333, -0.67850888, -0.54262607],
             [ -0.83333333,  0.92069052,  1.08553531],
             [ -0.66666667,  1.32099074,  0.68805757],
             [ -0.5       ,  0.91368526,  0.58832708],
             [ -0.33333333, -0.39629722, -0.34810409],
             [ -0.16666667,  1.17287966,  0.95260511],
             [  0.       ,  0.61245934,  0.77502344],
             [  0.16666667,  0.20115086,  0.27458273],
             [  0.33333333, -0.4433325 , -0.353265  ],
             [  0.5       , -0.3542657 , -0.29918013],
             [  0.66666667, -0.51838879, -0.37593514],
             [  0.83333333, -0.67850888, -0.54262607],
             [ -0.83333333,  0.32224168,  0.41947447],
             [ -0.66666667,  0.58043533,  0.57148107],
             [ -0.5       ,  0.32324243,  0.56526676],
             [ -0.33333333, -0.49737303, -0.43621894],
             [ -0.16666667,  0.46334751,  0.46471831],
             [  0.       ,  0.12309232,  0.1757552 ],
             [  0.16666667, -0.12309232,  0.03180267],
             [  0.33333333, -0.53540155, -0.42820137],
             [  0.5       , -0.47735802, -0.32256172],
             [  0.66666667, -0.57743307, -0.47901245],
             [  0.83333333, -0.67850888, -0.54262607],
             [ -0.83333333,  0.30122592, -0.03180267],
             [ -0.66666667,  0.49637228,  0.38411057],
             [ -0.5       ,  0.37628221,  0.27811973],
             [ -0.33333333, -0.5283963 , -0.50101858],
             [ -0.16666667,  0.32224168,  2.40143064],
             [  0.       ,  0.12709532,  1.08386749],
             [  0.16666667, -0.17513135, -0.26842423],

```

**Gambar 3.13.** Hasil *preprocessing* data