Analisis Permasalahan Perangkat Pencetak Menggunakan Metode Algoritma K-Means dan K-Medoids

Fadli Aziz Setiawan[1*], Mujiono Sadikin[2], Emil Robert Kaburuan[3]

[1,2,3] Program Studi Informatika, Universitas Mercu Buana, Jakarta Barat, DKI Jakarta
Email: [1*]41517120027@student.mercubuana.ac.id, [2]mujiono@dsn.ubharajaya.ac.id,
[3]emil.kaburuan@mercubuana.ac.id

(Naskah masuk: 15 Apr 2022, direvisi: 1 Jun 2022, diterima: 2 Jun 2022)

Abstrak

PT. Amido Makmor Tulus Sejati merupakan perusahaan distributor *multifunction printer* merek *Kyocera* di Indonesia. Evaluasi kinerja teknisi diperlukan untuk mempertahankan kepuasan *customer* terhadap penggunaan *multifunction printer Kyocera*. Proses penilaian kinerja teknisi masih dilakukan secara manual yang mengakibatkan hasil evaluasi kinerja teknisi yang diberikan kurang akurat atau kurang maksimal, sehingga perlu dilakukan suatu teknik pengolahan data secara cepat dan lebih akurat. Salah satunya dengan mempergunakan teknik data *mining* dengan menggunakan metode algoritma *clustering*. Metode algoritma *cluster*ing dipergunakan untuk mengelompokkan *problem* yang sering terjadi berdasarkan tipe mesin *multifunction printer Kyocera*. Pada penelitian ini diterapkan algoritma *clustering K-Means* dan *K-Medoids*, yang kemudian dilakukan uji *clustering* yang optimal dengan mempergunakan Metode *Elbow* dan *Silhouette Score*. Data yang dipergunakan dalam penelitian ini sebanyak 1.620 instan yang merupakan Data Kuantitatif. Proses untuk mencari nilai *clustering* yang optimal dilakukan dengan mencari rata-rata *Silhouette Score* dan Nilai Kemurnian dengan sisi luar dari algoritma *K-Means* dan *K-Medoids*. Hasil penelitian ini menunjukkan bahwa jumlah *cluster* optimal adalah 2 (dua) untuk algoritma *K-Means* dengan nilai *Silhouette Score* 0,606 dan jumlah *cluster* optimal 4 (empat) untuk algoritma *K-Medoids* dengan nilai *Silhouette Score* 0,240.

Kata Kunci: *Clustering, TF-IDF Vectorizer, Silhouette Score, K-Means, K-Medoids*

The Analysis of Printer Device Problem Using K-Means and K-Medoids Algorithm Method

Abstract

PT. Amido Makmor Tulus Sejati is a distributor of Kyocera brand multifunction printers in Indonesia. Evaluation of technician performance is needed to maintain customer satisfaction with the use of Kyocera multifunction printers. The process of evaluating the technicians performance is still done manually which the results of evaluating the technicians performance less accurate or less than optimal, so it is necessary to do a data processing technique quickly and more accurately. One of which is by using data mining techniques with the clustering algorithm method. The clustering algorithm method is used to group problems that often occur based on the type of Kyocera multifunction printer machine. In this study, the K-Means and K-Medoids clustering algorithms were applied, which was then tested for optimal clustering using the Elbow and Silhouette Score methods. The data used in this study were 1.620 instances which were quantitative data. The process to find the optimal clustering value is done by finding the average Silhouette Score and Purity Value with the outer side of the K-Means and K-Medoids algorithms. The results of this study indicate that the optimal number of clusters is 2 (two) for the K-Means algorithm with a Silhouette Score value of 0.606 and the optimal number of clusters is 4 (four) for the K-Medoids algorithm with a Silhouette Score of 0.240.

Keywords: *Clustering, TF-IDF Vectorizer, Silhouette Score, K-Means, K-Medoids*

I. PENDAHULUAN

Dalam segala bidang usaha, upaya peningkatan pengembangan usaha sangatlah penting. Pelanggan merupakan salah satu faktor utama dalam perkembangan perusahaan. Hubungan terhadap *customer* diperlukan untuk meningkatkan kualitas *customer* sehingga memberikan keuntungan yang lebih pada perusahaan [1].

Sebagai salah satu distributor penyedia *printer* merek *Kyocera* yang kegiatan bisnisnya melakukan penyewaan dan penjualan serta perbaikan kerusakan *printer* yang ada pada *customer*, perusahaan melakukan proses evaluasi kinerja teknisi yang masih belum sepenuhnya dilakukan secara maksimal dalam melakukan perbaikan kerusakan yang ada.

Sumber data penelitian ini berasal dari Data Panggilan *Customer* yang setiap harinya dibuat oleh admin perusahaan untuk dilakukan *cluster*ing data. Pada penelitian terkait sebelumnya, nilai k yang paling optimal untuk *k-means* adalah 3 dengan nilai rata-rata dalam jarak *centroid* (W) sebesar 35.241. Untuk *k-medoid*, nilai k yang paling optimal adalah 3 dengan nilai rata-rata dalam jarak *centroid* (W) sebesar 88.849 [2]. Penelitian lainnya, Nilai *Silhouette Score* pada proses *K-Means* yaitu mempunyai nilai sejumlah 0,558. Sementara Nilai *Silhouette Score* pada proses *K-Medoid* mempunyai nilai sejumlah 0,529, yang mengatakan proses K-Means menghasilkan nilai *Silhouette Score* lebih optimal dari K-Medoid. Maka dari itu *K-Means* dapat memberikan hasil pengelompokkan yang lebih optimal [3].

Pada penelitian ini data didapat dari admin perusahaan dalam rentang waktu April 2020 hingga Juni 2021. Metode yang dipergunakan yaitu Algoritma *K-Means* dan *K-Medoids* dengan *Metode Elbow* serta *Silhouette Score* dengan *TF-IDF Vectorizer* dan proses *Lower Case* untuk merapikan data yang tidak tersusun dengan baik.

II. LANDASAN TEORI

A. *Data Cleaning*

Data cleaning ialah menghilangkan atribut yang tidak digunakan serta kolom isian yang masih belum tersusun dengan baik seperti penggunaan huruf kapital dan kecil pada data yang tidak perlu digunakan untuk proses perhitungan nanti, agar data yang diolah sudah rapi dan sesuai [4].

B. Transformasi Data

Transformasi data adalah perubahan sejumlah *subdata* ke dalam bentuk yang sama agar memudahkan proses data *science*. Selanjutnya, melakukan perubahan data teks menjadi data numerik agar mempermudah penyajian serta pengolahan datanya [4].

C. TF-IDF

TF-IDF *vectorizer* merupakan proses penilaian kata dengan cara mengekstraksi ciri dari suatu teks. Terdiri dari 2 aspek: (1) *Term Frequency* (TF), frekuensi kemunculan istilah dalam dokumen; (2) *Inversed Document Frequency* (IDF), menilai pentingnya suatu *term*. Seluruh kata dianggap penting pada TF. IDF merupakan ukuran pentingnya suatu istilah yang diseimbangkan dengan frekuensi kemunculan istilah dalam kumpulan data.

D. Metode *Elbow*

Metode *Elbow* adalah proses untuk menentukan jumlah *cluster* yang optimal dengan membandingkan hasil antara jumlah *cluster* yang akan membuat sudut disuatu titik. Jika n *cluster* ke-1 dengan n *cluster* ke-2 menyampaikan sudut pada grafik atau nilainya mengalami penurunan yang besar, maka jumlah n *cluster* sudah benar. Dengan menghitung *Sum of Square Error* (SSE) dari *cluster* setiap nilai untuk mendapatkan perbandingan. Dikarenakan semakin besar jumlah K *cluster* maka nilai SSE akan semakin kecil. Rumus SSE sesuai dengan Persamaan 1.

$$SSE = \sum \sum |xi - ck \mid 2\ Xi\ K\ K{=}1 \quad (1)$$

Penjelasan:
K = *cluster* ke-c
xi = jarak data obyek ke-i
ck = pusat *cluster* ke-i [5].

E. Metode *Silhouette*

Metode Koefisien Siluet adalah kombinasi metode kohesi dan pemisahan. Koefisien Siluet ini sering dipakai untuk melihat tingkat dan kemampuan *cluster*, yaitu seberapa baik suatu obyek ditempatkan dalam *cluster*. Selanjutnya, dapat dipergunakan untuk menghitung seberapa dekat hubungan antar obyek dalam suatu *cluster*. Proses pemisahan yang berfungsi untuk menghitung seberapa jauh suatu *cluster* terpisah dari *cluster* lainnya.

Nilai yang diperoleh dari metode Koefisien Siluet terletak pada rentang nilai dari -1 sampai 1. Jika terdapat nilai Koefisien Siluet mendekati angka 1, maka kelompok dari obyek pada satu *cluster* akan semakin baik. Sedangkan nilai Koefisien Siluet mendekati angka -1, maka kelompok dari obyek pada satu *cluster* akan semakin buruk [6].

F. *Cluster*ing

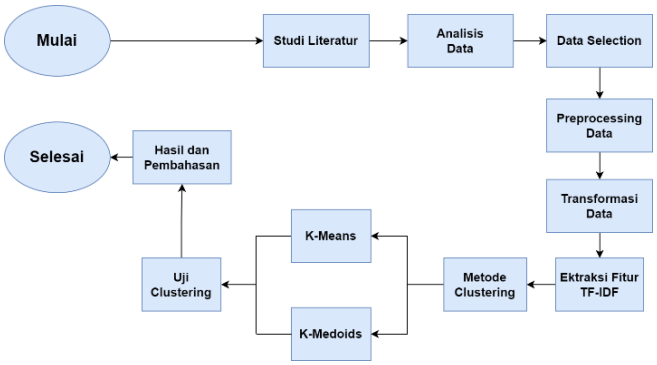
Clustering merupakan pengelompokkan sejumlah data atau obyek pada *cluster*, yang didalamnya berisi data yang serupa namun berbeda dari obyek *cluster* lainnya. *Clustering* adalah teknik pengelompokkan pada data *mining* yaitu mengelompokkan data maupun obyek pada suatu *cluster* sehingga masing-masing data pada *cluster* tersebut berisi suatu data yang memiliki kemiripan dengan obyek pada *cluster* lainnya. [7].

G. Data Numerik dan Data Kategorik

Data numerik adalah data kuantitatif yang nilainya berupa bilangan. Data kategorik adalah himpunan kategori dan masing-masing nilai mewakili sejumlah kategori. Data kategorikal sering disebut juga data kualitatif dengan bentuk tidak teratur [8].

III. METODOLOGI PENELITIAN

Dalam penelitian ini digunakan tahapan seperti disajikan pada Gambar 1.



Gambar. 1 Tahapan Penelitian

A. Studi Literatur

Pada proses ini, peneliti melakukan studi literatur dari berbagai sumber referensi seperti literatur berasal jurnal ilmiah serta karya ilmiah yang berkaitan dengan topik yang diteliti [4]. Lalu, dilanjutkan dengan pengumpulan data yang dibutuhkan serta diproses dalam pengerjaan data *science*. Sumber data yang dipergunakan dari PT. Amido Makmor Tulus Sejati yaitu data laporan panggilan *customer* setiap harinya. Data yang dicantumkan bersifat data mentah yang belum terseleksi menjadi data yang siap diolah pada penelitian.

B. Analisis Data

Proses selanjutnya yaitu analisis data dimana peneliti menganalisis data keseluruhan yang dimiliki dan diproses sesuai data yang sudah siap diolah untuk penelitian.

C. *Data Selection*

Membuat kumpulan data target, memilih kumpulan data, atau berfokus pada subset variabel dan sampel data, tempat penemuan dapat dilakukan. Hasil seleksi disimpan dalam sebuah *file*, terpisah dari *database* operasional [9].

Pada Tabel 1 dapat dilihat *selection* data awal untuk diolah ke tahap selanjutnya.

Tabel 1. *Data Selection* Awal

Cell	*Source Code*
1	# *Read Dataset* dengan 100 *rows* teratas
	data = pd.read_csv('C:/Users/Cango56/Documents/T.Informatika/Semester 8/Tugas Akhir/Dataset MPTI-TA(1).csv')
	data

Pada Tabel 2, merupakan *data selection* setelah diproses vektorisasi *TfidfVectorizer* menjadi Data Numerik.

Tabel 2. *Data Selection* Setelah Diubah Menjadi Numerik

Cell	*Source Code*
1	*##Data selection* setelah diubah menjadi Numerik
	data = data.copy()
	data = data[["Kategori_Problem", "Kategori_Type", "Kategori_Perbaikan", "Kategori_Status"]]
	data.head(100)

D. *Preprocessing Data*

Tahapan *preprocessing* data ini dilakukan proses data *cleaning*. *Cleaning* Data ialah menghilangkan atribut yang tidak digunakan serta kolom isian yang masih tidak tersusun dengan baik seperti penggunaan huruf kapital dan kecil pada data yang tidak dibutuhkan dalam proses perhitungan nantinya, sehingga data sudah rapi dan sesuai ketika siap diolah [4].

E. Transformasi Data

Selanjutnya, transformasi data adalah proses perubahan di sejumlah *subdata* menjadi satu bentuk yang sama untuk mempermudah proses *data science*. Selanjutnya, melakukan perubahan data teks menjadi data numerik untuk mempermudah dalam penyajian serta pengolahan datanya [4].

F. TF-IDF

TF-IDF dipergunakan dalam pembobotan kata dari beberapa kolom pada *dataset*. Rumus dari TF-IDF adalah:

$$\text{TF} = \frac{\text{Frekuensi term dalam satu dokumen}}{\text{Total kata dalam dalam satu dokumen}} \quad (2)$$

$$\text{IDF} = \log \frac{\text{Total dokumen}+1}{\text{Frekuensi dokumen mengandung term}} \quad (3)$$

$$TF - IDF = TF \times IDF \quad (4)$$

Pada penelitian ini memakai *library scikit-learn* pada penerapan TF-IDF untuk merubah data menjadi vektor.

G. *Clustering*

Terdapat dua metoda *cluster*ing yang dipergunakan:

1. Algoritma *K-Means*

Metode *K-Means* merupakan satu tipe *cluster*ing dengan konsep membagi n obyek atau data menjadi K *cluster*. *Output* dari metode K-Means ini bergantung pada jumlah K dan posisi titik *centroid* yang ditentukan di awal.

Proses untuk metode pengelompokan *K-Means*, seperti berikut:

- Masukan:
  K: jumlah *cluster*
  Ci: n *centroid* awal
  D: kumpulan data berisi n obyek
- Keluaran: Sebanyak set dari K *cluster*

- Tahapan:
  a. Input K *cluster*
  b. Input n *centroid* C untuk setiap *cluster*
  c. Menghitung jarak masing-masing data obyek ke masing-masing *centroid cluster*
  d. Posisikan data obyek ke *cluster* terdekat
  e. Mengubah n *centroid* pada rata-rata semua obyek untuk masing-masing *cluster*
  f. Mengulangi tahapan huruf c ke huruf e, sampai tidak ada data obyek yang bergerak di masing-masing K *cluster* [1].

2. Algoritma *K-Medoids*

Algoritma *K-Medoids* adalah algoritma *clustering* yang menyerupai *K-Means*. Perbandingan dari algoritma ini adalah memakai obyek sebagai representasi (*medoids*) menjadi pusat *cluster* untuk masing-masing *cluster*. Sedangkan *K-Means* menggunakan nilai rata-rata (*mean*) sebagai pusat dari gugus. *K-Medoids* mempunyai keunggulan dalam menangani kekurangan *K-Means* yang sensitif atas *noise* dan *outlier*. Obyek nilai yang besar mengharuskan untuk menyimpang dari distribusi data. Keuntungan lainnya adalah hasil dari proses *clustering* tidak terikat pada urutan *dataset* yang dimasukkan.

Berikut Tahapan algoritma K-Medoids:

1. Inisialisasi k pusat *cluster*(total *cluster*)
2. Berikan masing-masing data (obyek) ke *cluster* terdekat memakai persamaan standar *Euclidian Distance* dengan persamaan berikut:

$$d(x, y) = \|x - y\| = \sqrt{\sum} (xi - yi)\ 2\ n\ i{=}1\ ;\ 1{,}2{,}3{,}\ \ldots\ n \qquad (5)$$

3. Memilih obyek secara acak di setiap *cluster* sebagai calon *medoid* baru.
4. Menghitung jarak masing-masing obyek yang ada di setiap *cluster* dengan calon *medoid* baru.
5. Menghitung jumlah simpangan (S) dengan menghitung nilai jarak total baru – jarak total lama. Jika S < 0, maka *swap* obyek oleh data *cluster* untuk membentuk satu set baru k obyek baru sebagai *medoid*.
6. Mengulangi tahapan no 3 sampai 5 sampai tidak ada perubahan *medoid*, sehingga diperoleh *cluster* dan anggota *cluster*-nya masing-masing [10].

IV. HASIL DAN PEMBAHASAN

Berdasarkan Gambar 1 menunjukkan data yang didapat dari perusahaan PT. Amido Makmor Tulus Sejati dengan rentang waktu Juni 2020 – Juni 2021 dengan 13 kolom dan 1.620 baris masih berupa data mentah.

	No	Customer	Type	S/N	STATUS	Date Call	Time Call	Time In	Time Out	Respon Time(Menit)	Down Time(Menit)	PROBLEM	PERBAIKAN
0	1	Bank Victoria Cab. Mangga Besar	M-2535dn	LZP5X03812	GSP	6/8/2020	12:15 PM	12:49 PM	1:40 PM	34	85	Mesin noise	GSP Heatroll dan Bushing
1	2	Kementerian Pekerja Umum Bag. Umum	TA-3501i	LUN3Z00240	UMC	6/3/2020	9:20 AM	9:58 AM	10:51 AM	38	91	Hasil Bentol dan Garis	Clean MC dan Clean Sit Glass
2	3	Bank Panin Cab. Sudirman Bag. Teller	M-2535dn	LZP6Z08461	UMC	6/14/2020	12:02 PM	12:42 PM	1:49 PM	40	107	Paper Jam	Setting Cassete Folio
3	4	Metro 55	TA-1800	LT14Y16405	UMC	6/5/2020	8:30 AM	9:06 AM	10:05 AM	36	95	Hasil Bergaris	Kuras Drum, Clean Cleaning Blade
4	5	Bank OCBC NISP Cab. Tendean	M-2535dn	LZP6405812	UMC	6/3/2020	11:30 AM	12:47 PM	1:38 PM	77	128	Paper Jam DP + Cassete	Clean Pulley DP, Clean Pulley Cassete
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1615	1616	Halik Selindo Alpha	TA-180	QLZ1306551	UMC + MM	6/25/2021	12:30 PM	1:40 PM	2:20 PM	70	110	Hasil bergaris	Maintenance, Clean shit glass
1616	1617	Halik Selindo Alpha	TA-180	QLZ1Z09159	UMC + MM	6/28/2021	3:00 PM	4:10 PM	4:50 PM	70	110	Hasil bergaris	Maintenance, Clean shit glass
1617	1618	Halik Selindo Alpha	TA-180	QLZ0103834	UMC + MM	6/28/2021	10:00 AM	10:50 AM	11:30 AM	50	90	Hasil bergaris	Maintenance, Clean shit glass
1618	1619	Halik Selindo Alpha	M-2540dn	LZP7610230	UMC	6/29/2021	11:00 AM	11:50 AM	12:30 PM	50	90	Hasil kotor	Kuras drum, Clean CL, Order drum+Cleaning blade
1619	1620	PT. Kuratayeso	M-2540dn	VYA8804705	UMC	6/30/2021	4:00 PM	4:37 PM	5:36 PM	37	96	Hasil kotor	Clean MC

1620 rows × 13 columns

Gambar 2. Membaca *Dataset* Pada CSV

Setelah diketahui data mentah yang didapat dari perusahaan PT. Amido Makmor Tulus Sejati berdasarkan pada Gambar 2, Peneliti melakukan proses seleksi data awal untuk diolah selanjutnya pada Gambar 3 di bawah ini.

```
## Data Selection terhadap beberapa column
data = data[["Customer", "Type", "S/N", "STATUS", "PROBLEM", "PERBAIKAN"]]
data
```

	Customer	Type	S/N	STATUS	PROBLEM	PERBAIKAN
0	Bank Victoria Cab. Mangga Besar	M-2535dn	LZP5X03812	GSP	Mesin noise	GSP Heatroll dan Bushing
1	Kementerian Pekerja Umum Bag. Umum	TA-3501i	LUN3Z00240	UMC	Hasil Bentol dan Garis	Clean MC dan Clean Sit Glass
2	Bank Panin Cab. Sudirman Bag. Teller	M-2535dn	LZP6Z08461	UMC	Paper Jam	Setting Cassete Folio
3	Metro 55	TA-1800	LT14Y16405	UMC	Hasil Bergaris	Kuras Drum, Clean Cleaning Blade
4	Bank OCBC NISP Cab. Tendean	M-2535dn	LZP6405812	UMC	Paper Jam DP + Cassete	Clean Pulley DP, Clean Pulley Cassete
...	...	...	...	...	...	...
1615	Halik Selindo Alpha	TA-180	QLZ1306551	UMC + MM	Hasil bergaris	Maintenance, Clean shit glass
1616	Halik Selindo Alpha	TA-180	QLZ1Z09159	UMC + MM	Hasil bergaris	Maintenance, Clean shit glass
1617	Halik Selindo Alpha	TA-180	QLZ0103834	UMC + MM	Hasil bergaris	Maintenance, Clean shit glass
1618	Halik Selindo Alpha	M-2540dn	LZP7610230	UMC	Hasil kotor	Kuras drum, Clean CL, Order drum+Cleaning blade
1619	PT. Kuratayeso	M-2540dn	VYA8804705	UMC	Hasil kotor	Clean MC

1620 rows × 6 columns

Gambar 3. Data *Selection* Awal

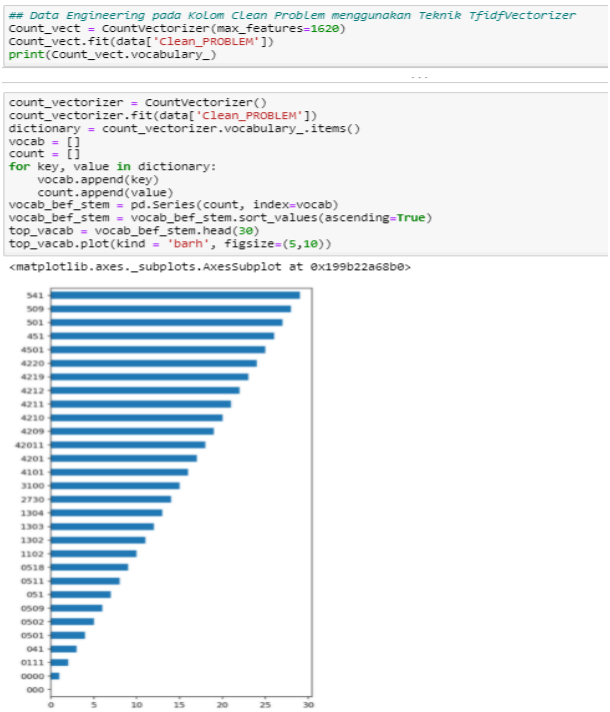
Berdasarkan pada Gambar 3 proses seleksi data di atas, Tahapan selanjutnya dilakukan proses *preprocessing lower case* dan menghilangkan karakter serta tanda baca yang tidak diperlukan pada kolom *Problem* dan Perbaikan seperti pada Tabel 3.

Tabel 3. *Cleaning Text* Pada Kolom *Problem* dan Perbaikan

Cell	*Source Code*
1	*## Data Preprocessing*
	import re
	import nltk
	from bs4 import BeautifulSoup
	tok = WordPunctTokenizer()
	pat1 = r'@[A-Za-z0-9_]+'
	pat2 = r'https?://[^]+'
	combined_pat = r'|'.join((pat1, pat2))
	www_pat = r'www.[^]+'
	#set_stopword yang di deskripsikan sendiri
	stopword_user=set(pd.read_csv('C:/Users/Cango 56/Documents/T.Informatika/Semester8/Tu

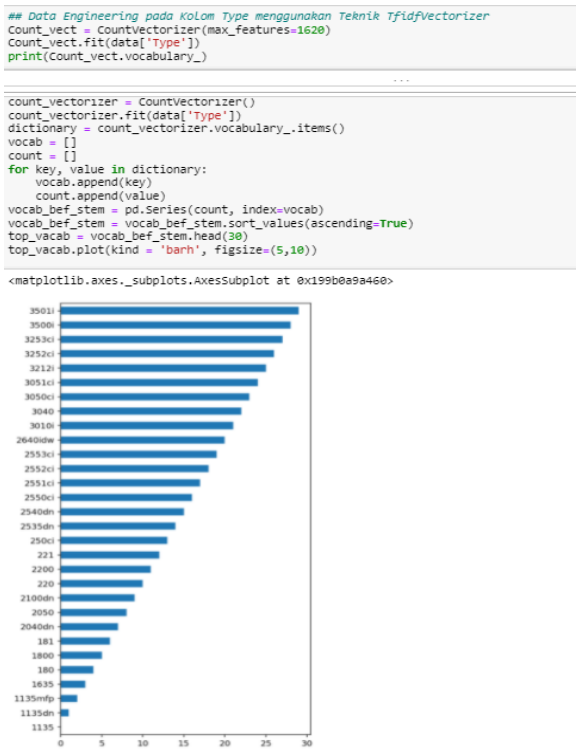
	gasAkhir/DatasetMPTI-TA(1).csv', sep='\n', header=0))
2	def proses_teks(teks):
	soup = BeautifulSoup(teks, 'lxml')
	souped = soup.get_text()
	try:
	teks = souped.decode("utf-8-sig").replace(u"\ufffd", "?")
	except:
	teks = souped
	teks_bersih= re.sub("[^a-zA-Z0-9]", " ",(re.sub(www_pat, '', re.sub(combined_pat, '', teks)).lower()))
	teks_bersih= ' '.join([word for word in teks_bersih.split() if word not in stopword_user])
	return (" ".join([x for x in tok.tokenize(teks_bersih) if len(x) > 1])).strip()
3	x=[]
	for teks in data.PROBLEM: x.append(proses_teks(teks))
4	clean_text=pd.DataFrame({'Clean_PROBLEM': x})
5	data=pd.concat([data,clean_text],axis=1)
	data

Berdasarkan Gambar 4, memakai *library Scikit-learn, Class TfidfVectorizer* dapat menghitung nilai TF-IDF di kolom *Clean_Problem* dan mengubahnya menjadi bentuk *vektor*.



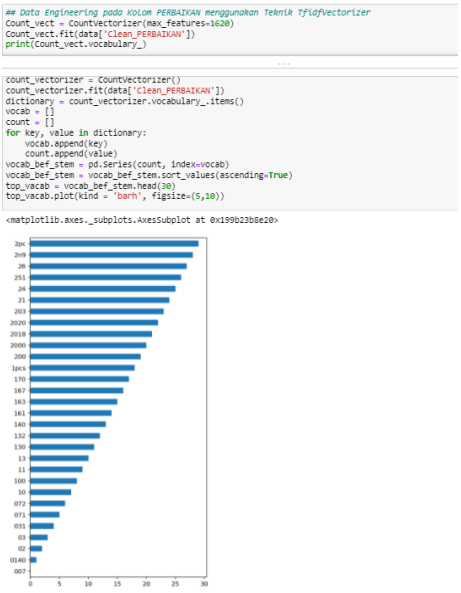
Gambar 4. Proses *TF-IDF Vectorizer* Pada Kolom *Clean_Problem*

Berdasarkan Gambar 5, memakai *library Scikit-learn, Class TfidfVectorizer* dapat menghitung nilai TF-IDF pada kolom *Type* dan mengubahnya ke dalam bentuk *vektor*.



Gambar 5. Proses *TF-IDF Vectorizer* Pada Kolom *Type*

Berdasarkan Gambar 6, memakai *library Scikit-learn, Class TfidfVectorizer* dapat menghitung nilai TF-IDF pada kolom *Clean_Perbaikan* dan mengubahnya menjadi bentuk *vektor*.



Gambar 6. Proses *TF-IDF Vectorizer* Pada Kolom *Clean_Perbaikan*

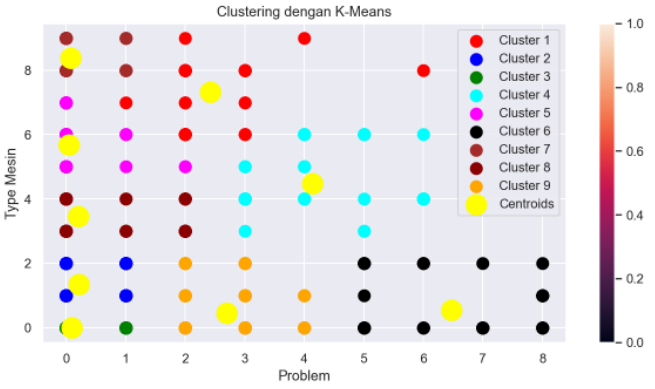
Berdasarkan pada Gambar 7, dilakukan proses seleksi data kembali setelah data sudah diubah menjadi numerik.

	Kategori_Problem	Kategori_Type	Kategori_Perbaikan	Kategori_Status
0	5	2	0	4
1	0	0	0	1
2	1	2	0	1
3	2	7	0	1
4	0	2	0	1
...	...	...	...	...
95	0	0	0	1
96	0	8	2	5
97	0	9	2	5
98	0	0	0	1
99	0	9	0	6

100 rows × 4 columns

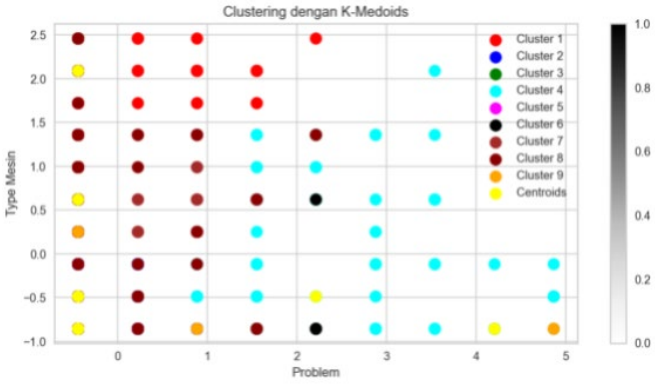
Gambar 7. *Data Selection* Setelah Diubah Menjadi Numerik

Berdasarkan Gambar 8, dilakukan uji *clustering* awal menggunakan metode algoritma *K-Means* dengan nilai *cluster*ing awal = 9 setelah didapat *labels cluster* tiap barisnya.



Gambar 8. Algoritma *K-Means* Awal *n_cluster* = 9

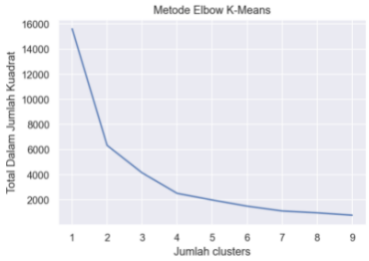
Berdasarkan Gambar 9, dilakukan uji *clustering* awal menggunakan metode algoritma *K-Medoids* dengan nilai *clustering* awal = 9 setelah didapat labels *cluster* tiap barisnya.



Gambar 9. Algoritma *K-Medoids* Awal *n_cluster* = 9

Berdasarkan Gambar 10, setelah dilakukan *cluster*ing awal dengan *n_cluster* = 9 pada algoritma K-Means, dilakukan proses Metode Elbow pada K-Means untuk diketahui nilai *cluster* berapa yang optimal (tepat) dengan bentuk siku terletak pada K = 2.

```
# Menggunakan Metode Elbow untuk menentukan angka cluster yang tepat
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 10):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter=100, random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 10), wcss)
plt.title('Metode Elbow K-Means')
plt.xlabel('Jumlah clusters')
plt.ylabel('Total Dalam Jumlah Kuadrat')
plt.show()
```



Gambar 10. Metode *Elbow K-Means*

Berdasarkan Gambar 11, setelah dilakukan proses Metode Elbow selanjutnya dapat dilihat nilai Rata-Rata *Silhouette Score* pada algoritma K-Means yaitu nilai 0,606.

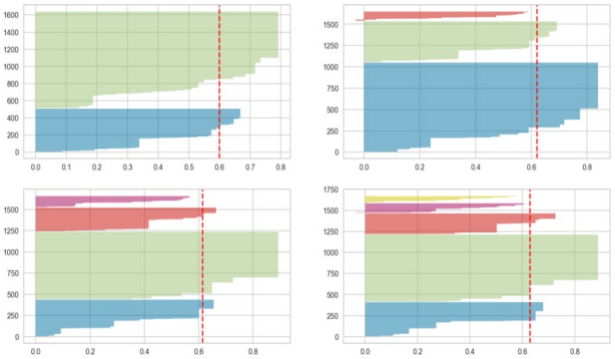
```
## Mengetahui nilai rata-rata Silhouette pada K-Means
silhouette_avg = silhouette_score(X, kmeans.labels_, metric='euclidean')
print('Silhouetter Score: %.3f' % silhouette_avg)
```

```
Silhouetter Score: 0.606
```

Gambar 11. Rata-Rata *Silhouette Score* K-Means

Berdasarkan Gambar 12, setelah didapat nilai Rata-Rata *Silhouette Score* pada Algoritma *K-Means* selanjutnya dibuat visualisasi dari *Silhouette Score* tersebut menggunakan *library Yellowbrick*.

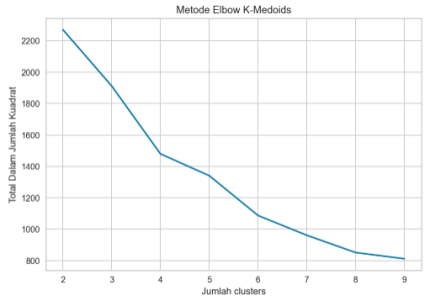
```
from yellowbrick.cluster import SilhouetteVisualizer

fig, ax = plt.subplots(2, 2, figsize=(15,8))
for i in [2, 3, 4, 5]:
    '''
    Buat instance KMeans untuk jumlah cluster yang berbeda
    '''
    kmeans = KMeans(n_clusters=i, init='k-means++', n_init=10, max_iter=100, random_state=42)
    q, mod = divmod(i, 2)
    '''
    Buat instance SilhouetteVisualizer dengan instance KMeans
    Sesuaikan visualisatornya
    '''
    visualizer = SilhouetteVisualizer(kmeans, colors='yellowbrick', ax=ax[q-1][mod])
    visualizer.fit(X)
```



Gambar 12. Visualisasi *Silhouette Score K-Means*

Berdasarkan Gambar 13, setelah dilakukan *clustering* awal dengan *n_cluster* = 9 pada algoritma *K-Medoids*, dilakukan proses Metode *Elbow* pada *K-Medoids* untuk diketahui nilai *cluster* berapa yang optimal (tepat) dengan bentuk siku terletak pada K = 4.

```
## Menggunakan Silhouette Width untuk menemukan jumlah cluster
from sklearn_extra.cluster import KMedoids
sw = []
for i in range(2, 10):
    kmedoids = KMedoids(n_clusters = i, init = 'k-medoids++', max_iter=100, random_state = 42)
    kmedoids.fit(X)
    sw.append(kmedoids.inertia_)
plt.plot(range(2, 10), sw)
plt.title('Metode Elbow K-Medoids')
plt.xlabel('Jumlah clusters')
plt.ylabel('Total Dalam Jumlah Kuadrat')
plt.show()
```



Gambar 13. Metode *Elbow K-Medoids*

Berdasarkan Gambar 14, Setelah dilakukan proses Metode *Elbow* selanjutnya dapat dilihat nilai rata-rata *Silhouette Score* pada algoritma *K-Medoids* yaitu nilai 0,240.

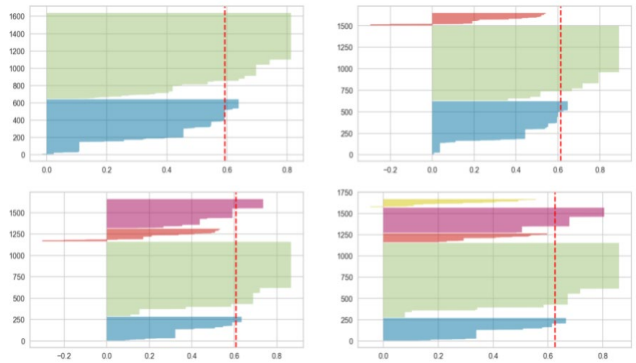
```
## Mengetahui nilai rata-rata Silhouette pada K-Medoids
silhouette_avg = silhouette_score(X, kMedoids.labels_, metric='euclidean')
print('Silhouetter Score: %.3f' % silhouette_avg)
```

```
Silhouetter Score: 0.240
```

Gambar 14. Rata-Rata *Silhouette Score K-Medoids*

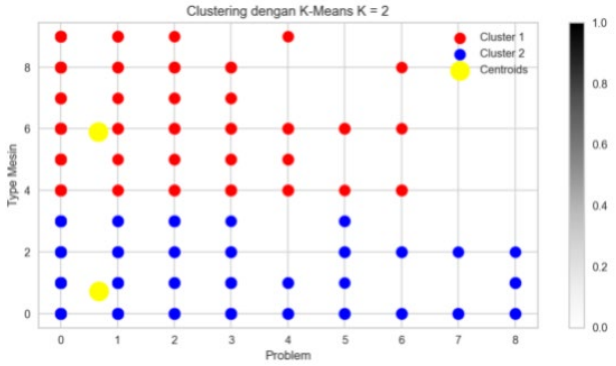
Berdasarkan Gambar 15, Setelah didapat nilai Rata-Rata *Silhouette Score* pada Algoritma K-Medoids selanjutnya dibuat visualisasi dari *Silhouette Score* tersebut menggunakan *library Yellowbrick.*

```
from yellowbrick.cluster import SilhouetteVisualizer

fig, ax = plt.subplots(2, 2, figsize=(15,8))
for i in [2, 3, 4, 5]:
    '''
    Buat instance KMedoids untuk jumlah cluster yang berbeda
    '''
    kmedoids = KMedoids(n_clusters = i, init = 'k-medoids++', max_iter=100, random_state=42)
    q, mod = divmod(i, 2)
    '''
    Buat instance SilhouetteVisualizer dengan instance KMedoids
    Sesuaikan visualisatornya
    '''
    visualizer = SilhouetteVisualizer(kmedoids, colors='yellowbrick', ax=ax[q-1][mod])
    visualizer.fit(X)
```



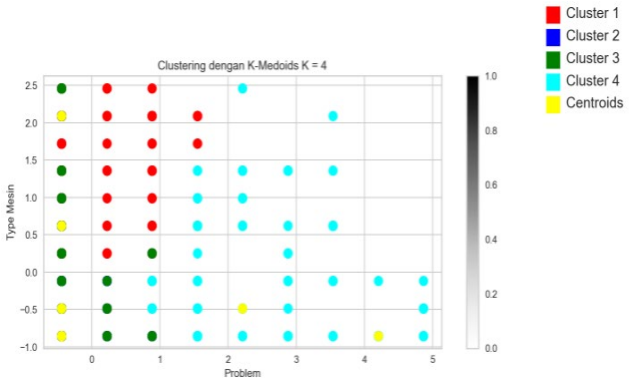
Gambar 15. Visualisasi *Silhouette Score K-Medoids*

Berdasarkan Gambar 16, merupakan visualisasi *clustering* dengan *K-Means* yang optimal yaitu K = 2 setelah diketahui Metode *Elbow* dan Nilai Rata-Rata *Silhouette Score*-nya. Dari hasil visualisasi yang didapat, kategori tipe 1 dan 6 menunjukkan bahwa untuk Tipe Mesin "M-2540dn dan FS-6525MFP" sering mengalami kerusakan "*Paper Jam*" pada Kategori Problem 1.



Gambar 16. Algoritma *K-Means* Optimal K = 2

Berdasarkan Gambar 17, merupakan visualisasi *clustering* dengan *K-Medoids* yang optimal yaitu K = 4 setelah diketahui metode *Elbow* dan nilai rata-rata *Silhouette Score*-nya. Dari Hasil Visualisasi yang didapat, Kategori 1, 2, 4 dan 7 menunjukkan bahwa untuk Tipe Mesin "M-2540dn, M-2535dn, TA-2551ci, dan TA-1800" sering mengalami kerusakan "*Paper Jam, Mesin Noise* dan *Error C7990*" pada Kategori Problem 1, 5 dan 9.



Gambar 17. Algoritma *K-Medoids* Optimal K = 4

V. KESIMPULAN DAN SARAN

Berdasarkan hasil percobaan serta pembahasan sudah dilakukan tahapan proses studi literatur, analisis data, *data selection, preprocessing data*, transformasi data. Selanjutnya, uji metode *clustering* dengan TF-IDF *Vectorizer* untuk pembobotan kata dan mengubahnya ke dalam bentuk vektor. Percobaan algoritma *K-Means* menghasilkan nilai *cluster* yang optimal atau tepat pada K = 2 dengan rata–rata *Silhouette Score* sebesar 0,606. Sedangkan untuk percobaan algoritma *K-Medoids* menghasilkan nilai *cluster* yang optimal atau tepat pada K = 4 dengan rata–rata *Silhouette Score* sebesar 0,240

dapat disimpulkan untuk metode algoritma *clustering* terbaik pada dataset ini adalah *K-Means*. Karena untuk nilai *centroid*-nya terdapat 2 kategori dengan nilai *cluster* yang optimal pada visualisasinya dengan menunjukkan bahwa tipe mesin "M-2540dn dan FS-6525MFP" sering mengalami kerusakan "*Paper Jam*".

Dengan hasil penelitian ini, perusahaan bisa mengambil keputusan kerusakan apa saja yang sering terjadi berdasarkan tipe mesin yang ada pada *customer* supaya tidak timbul kekecewaan *customer* terhadap mesin yang digunakan.

Untuk penelitian selanjutnya, penulis menyarankan untuk memakai data uji yang lebih banyak supaya nilai *cluster* yang optimal bisa lebih optimal. Setelah itu, dapat dilakukan dengan modifikasi metoda *clustering* yang lebih optimal.

REFERENSI

[1] S. A. Sutresno, A. Iriani, and E. Sediyono, "Metode K-Means Clustering dengan Atribut RFM untuk Mempertahankan Pelanggan," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 4, no. 3, pp. 433–440, Dec. 2018.

[2] M. Aryuni, E. D. Madyatmadja, and E. Miranda, "Penerapan K-Means dan K-Medoids Clustering Pada Data Internet Banking di Bank XYZ," *Jurnal Teknik dan Ilmu Komputer*, vol. 7, no. 27, pp. 349–356, Jan. 2018.

[3] Athifaturrofifah, R. Goejantoro, and D. Yuniarti, " Perbandingan Pengelompokan K-Means dan K-Medoids Pada Data Potensi Kebakaran Hutan/Lahan Berdasarkan Persebaran Titik Panas (Studi Kasus : Data Titik Panas Di Indonesia Pada 28 April 2018) ," *Jurnal Eksponensial*, vol. 10, no. 2, pp. 143–152, 2019.

[4] A. Supriyadi, A. Triayudi, and I. D. Sholihati, "Perbandingan Algoritma K-Means Dengan K-Medoids Pada Pengelompokan Armada Kendaraan Truk Berdasarkan Produktivitas," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 6, no. 2, pp. 229–240, 2021.

[5] D. A. Dewi and D. A. Pramita, "Analisis Perbandingan Metode Elbow dan Silhouette Pada Algoritma Clustering K-Medoids Dalam Pengelompokan Produksi Kerajinan Bali," *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 9, no. 3, pp. 102–109, 2019.

[6] G. R. Prima, "Analisa Perbandingan Nilai K Terbaik Untuk Clustering K-Means Menggunakan Pendekatan Elbow dan Silhouette Pada Citra Aksara Jawa," Universitas Sanata Dharma, 2021.

[7] A. Nofiar, S. Defit, and Sumijan, "Penentuan Mutu Kelapa Sawit Menggunakan Metode K-Means Clustering," *Jurnal KomtekInfo*, vol. 5, no. 3, pp. 1–9, 2019.

[8] H. Putri and D. Saputro, "Clustering Data Campuran Numerik dan Kategorik Menggunakan Algoritme Ensemble Quick RObust Clustering using linKs (QROCK)", *Prisma*, vol. 5, pp. 716-720, Feb. 2022.

[9] M. Silalahi, "Analisis Clustering Menggunakan Algoritma K-Means Terhadap Penjualan Produk Pada PT Batamas Niaga Jaya", *CBIS*, vol. 6, no. 2, pp. 20–35, Sep. 2018.

[10] D. F. Pramesti, M. T. Furqon, and C. Dewi, "Implementasi Metode K-Medoids Clustering Untuk Pengelompokan Data Potensi Kebakaran Hutan/Lahan Berdasarkan Persebaran Titik Panas (Hotspot) ," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 9, pp. 723–732, Jun. 2017.