

Nama : Agil Rahmat

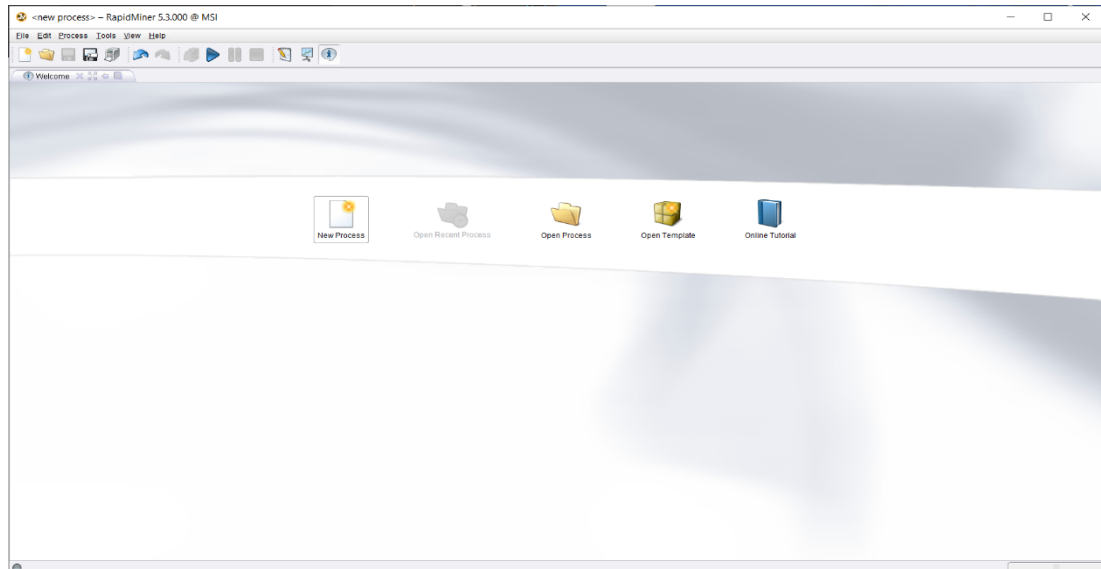
NIM : 2106037

Informatika A

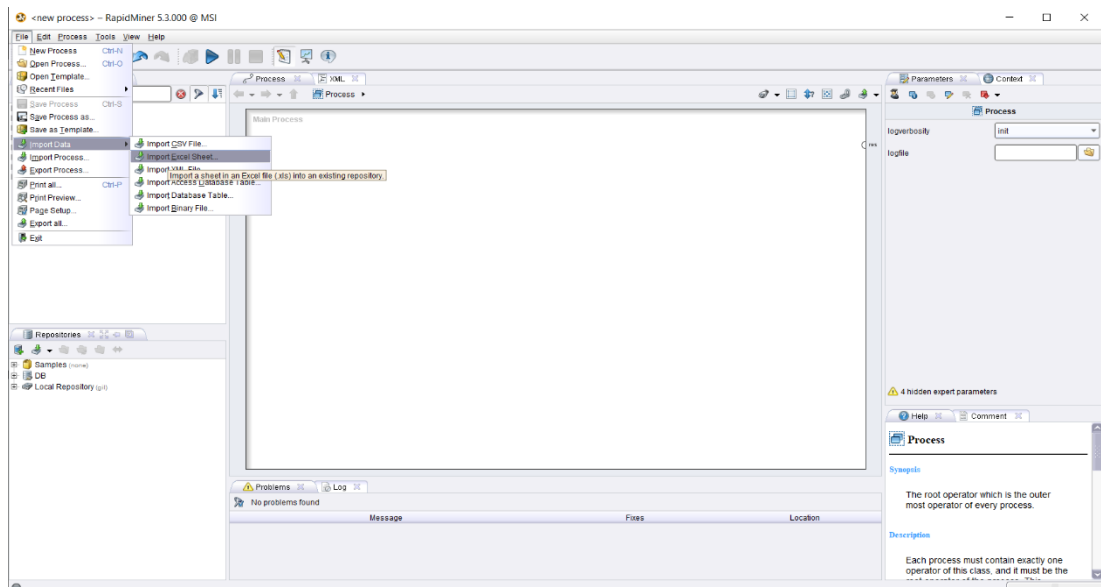
16 Mei 2023

1. Tahapan Machine Learning Menggunakan Rapid Minner.

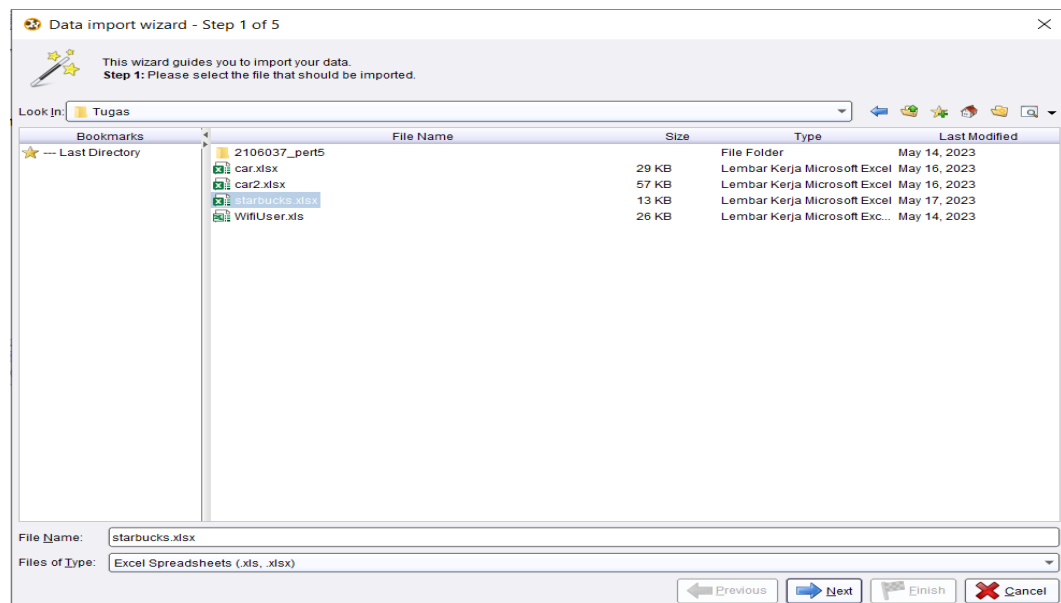
1) Buka RapidMiner Studio dan buatlah sebuah proses baru. Pilih New Process.



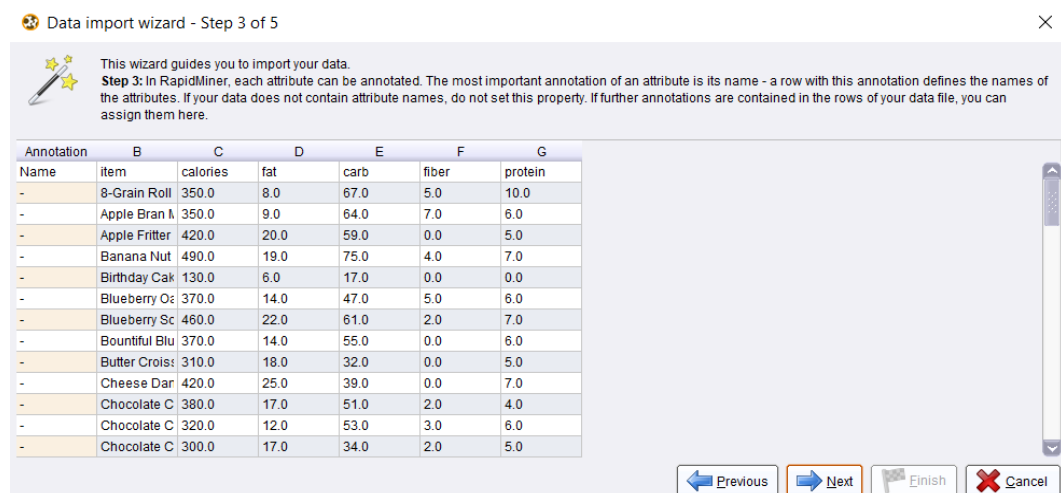
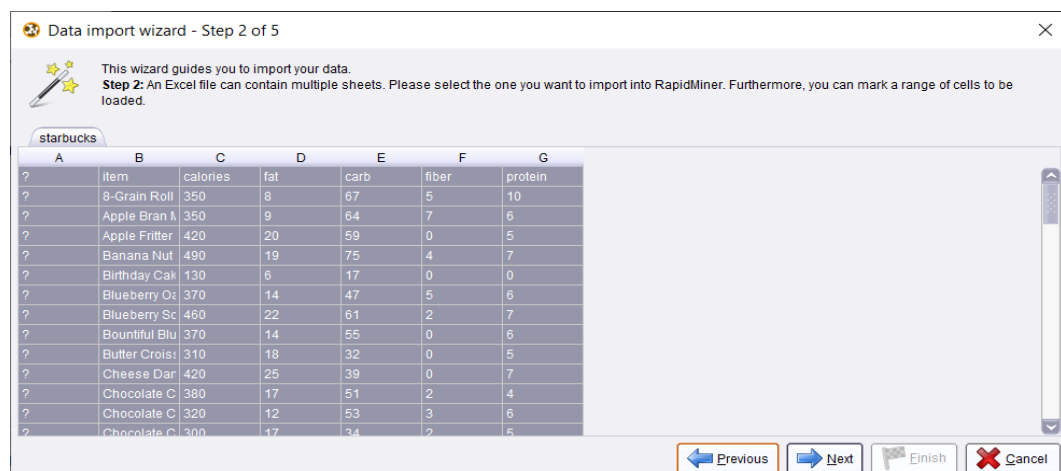
2) Klik menu File, lalu klik sub menu Import Data disitu ada beberapa pilihan untuk mengimport data berdasarkan ekstensi dari data. Pilih Import Excel Sheet.



- 3) Selanjutnya, pilih dataset yang ingin diimpor dengan menavigasi ke folder lokasi file dataset tersebut. Lalu klik Next.



- 4) Setelah dataset terimpor, RapidMiner akan membuka "Metada" editor yang menampilkan informasi dataset. Klik Next.



- 5) Selanjutnya mengidentifikasi type data dari atribut dataset yang diimpor apakah itu polynomial, binominal, integer dan sebagainya dan ganti kolom item atribut menjadi label. Setelah itu klik Next

Data import wizard - Step 4 of 5

This wizard guides you to import your data.
Step 4: RapidMiner uses strongly typed attributes. In this step, you can define the data types of your attributes. Furthermore, RapidMiner assigns roles to the attributes, defining what they can be used for by the individual operators. These roles can be also defined here. Finally, you can rename attributes or deselect them entirely.

☒ Reload data ☒ Guess value types ☒ Preview uses only first 100 rows. Date format: [Date format]

item	calories	fat	carb	fiber	protein
polynomial	integer	real	integer	integer	integer
label	attribute	attribute	attribute	attribute	attribute
8-Grain Roll	350	8.0	67	5	10
Apple Bran M.	350	9.0	64	7	6
Apple Fritter	420	20.0	59	0	5
Banana Nut	490	19.0	75	4	7
Birthday Cak	130	6.0	17	0	0
Blueberry O	370	14.0	47	5	6
Blueberry Sc	460	22.0	61	2	7

0 errors. ☒ Ignore errors ☐ Show only errors

Row, Column Error Original value Message

- 6) Terakhir simpan dataset yang diimpor pada local repository dalam file data serta beri nama dataset tersebut. Lalu klik Finish.

Data import wizard - Step 5 of 5

This wizard guides you to import your data.
Step 5: Please specify a repository location.

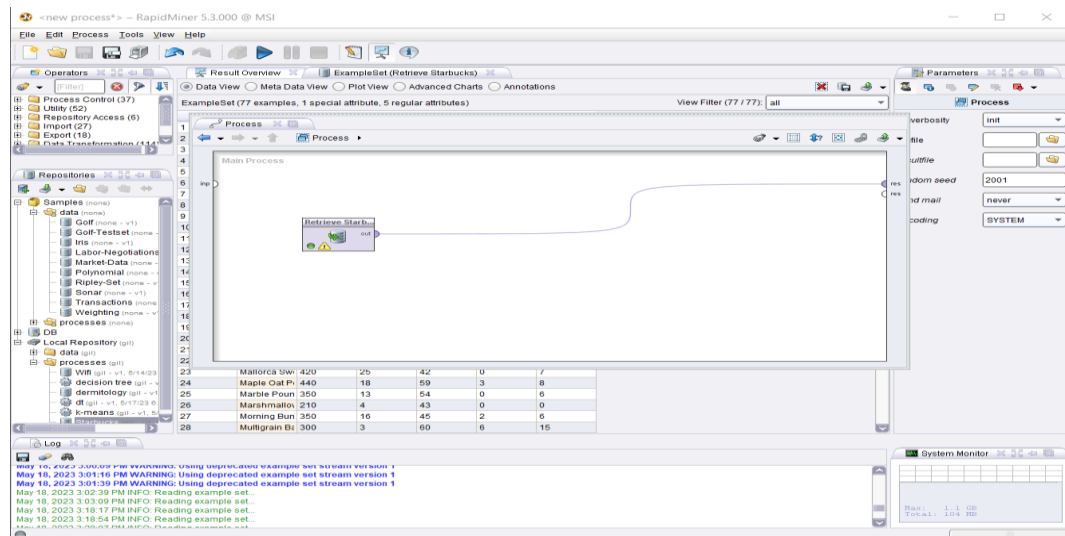
Local Repository (gill)

- data (gill)
- processes (gill)
 - Wifi (gill - v1, 5/14/23 7:30 PM - 1 kB)
 - decision tree (gill - v1, 5/16/23 9:07 PM - 1 kB)
 - dermatology (gill - v1, 5/18/23 2:42 PM - 53 kB)
 - dt (gill - v1, 5/17/23 6:55 PM - 3 kB)
 - k-means (gill - v1, 5/17/23 7:53 PM - 3 kB)

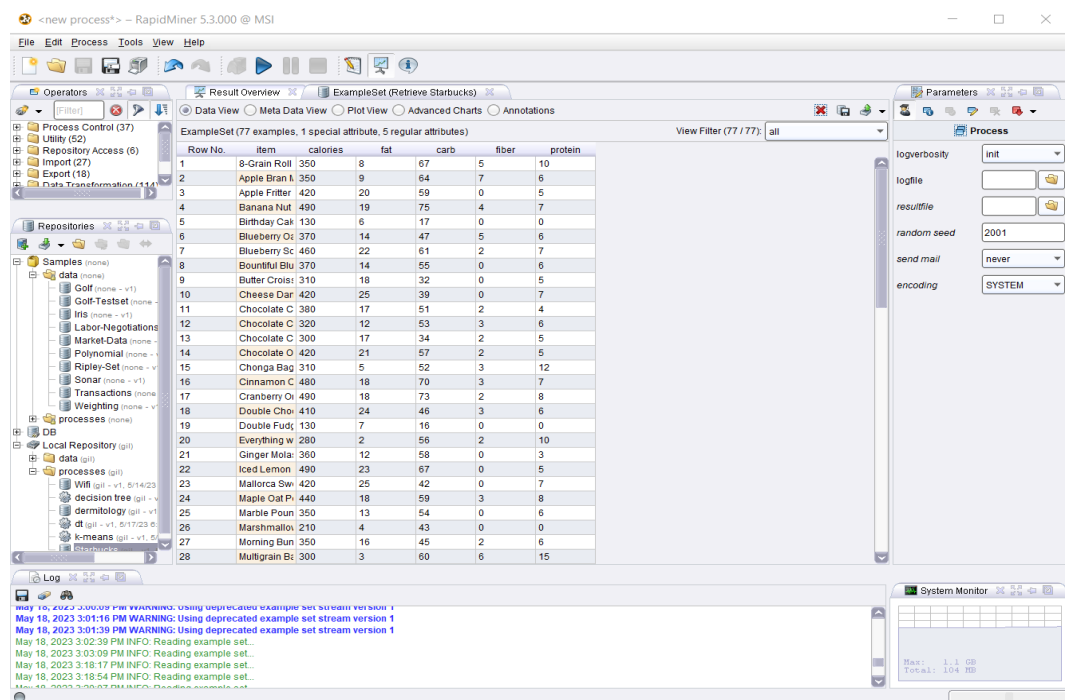
Name: Starbucks

Location: //Local Repository/processes/Starbucks

- 7) Selanjutnya, ke menu repositories yang ada di sebelah kiri bawah, buka Samples -> data -> car lalu tarik Iris ke workspace sehingga tampil seperti berikut.



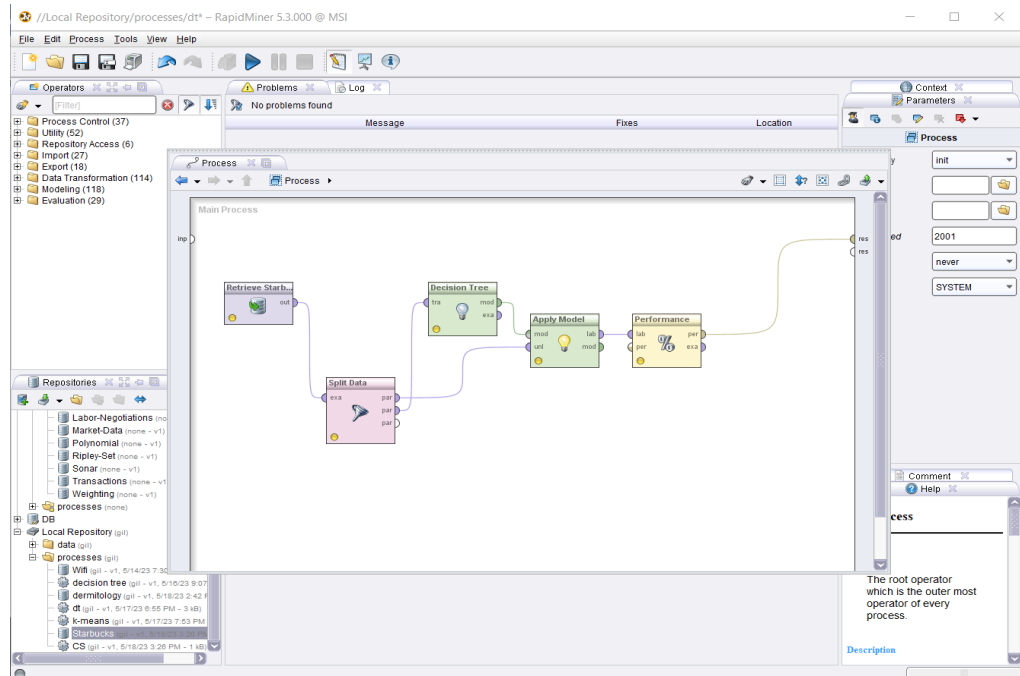
- 8) Jika sudah, maka akan tampil seperti berikut.



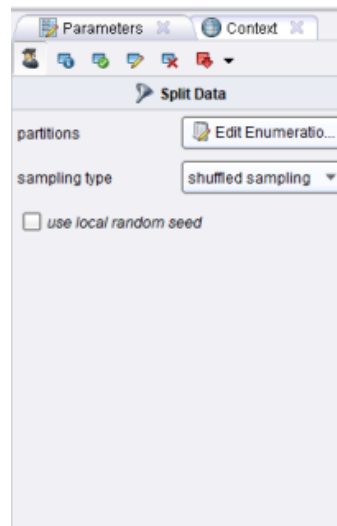
- 9) Selanjutnya melihat dataset tersebut dan ternyata dataset tidak memiliki missing value dan noise sehingga dapat melanjutkan ke tahapan selanjutnya yaitu pelatihan (Training).

10) Decision Tree

- a) Drag and drop operator Decision Tree, Set Role (2 buah), Apply Model, Split Data dan Performance dari panel operator di sebelah kiri ke proses di workspace sehingga terlihat seperti berikut.



- b) Lalu, klik operator Split Data sehingga di menu parameters terlihat seperti berikut.

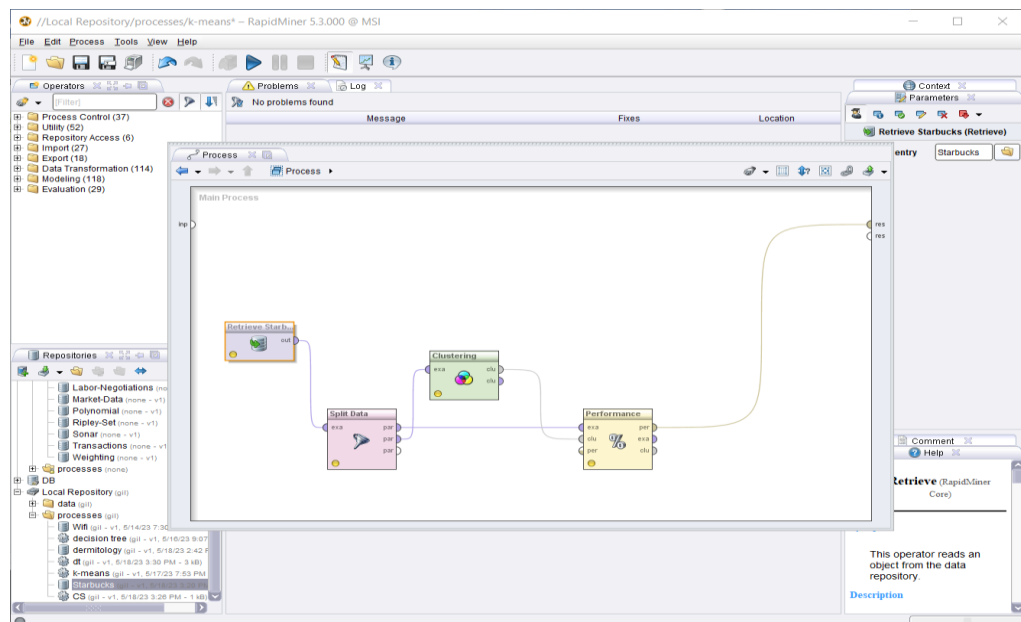


- c) Setelah itu klik Edit Enumeration untuk menentukan dataset ini akan dibagi menjadi berapa persen untuk data training dan data testingnya. Untuk kali ini, dataset akan dibagi 70% untuk data training dan 30% untuk data testing, maka buat entry seperti berikut.

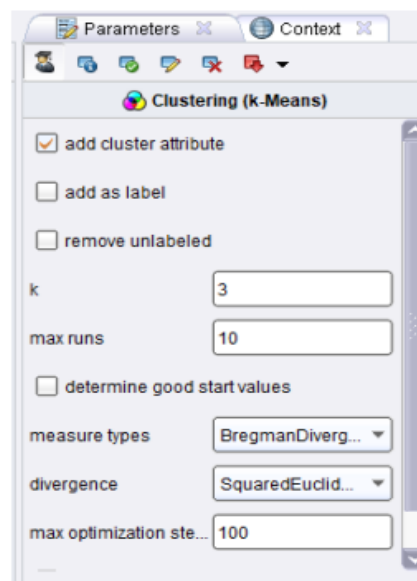
- d) Setelah itu jalankan dan hasilnya akan tampil seperti berikut.

11) K-Means

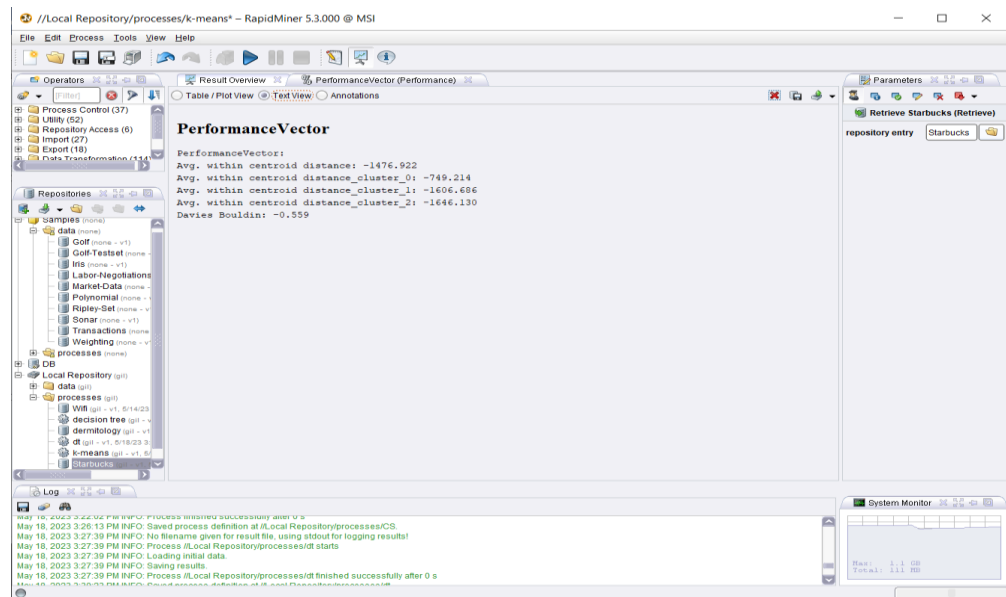
- a) Untuk kasus clustering, kita akan menggunakan dataset yang sama dengan yang sebelumnya yaitu dataset bunga Iris. Untuk melakukannya, hapus operator Decision Tree, Apply Model dan Performance. Setelah itu, tarik dan letakkan operator K-Means dan Cluster Distance Performance ke workspace dan sambungkan antar operatornya sehingga seperti berikut.



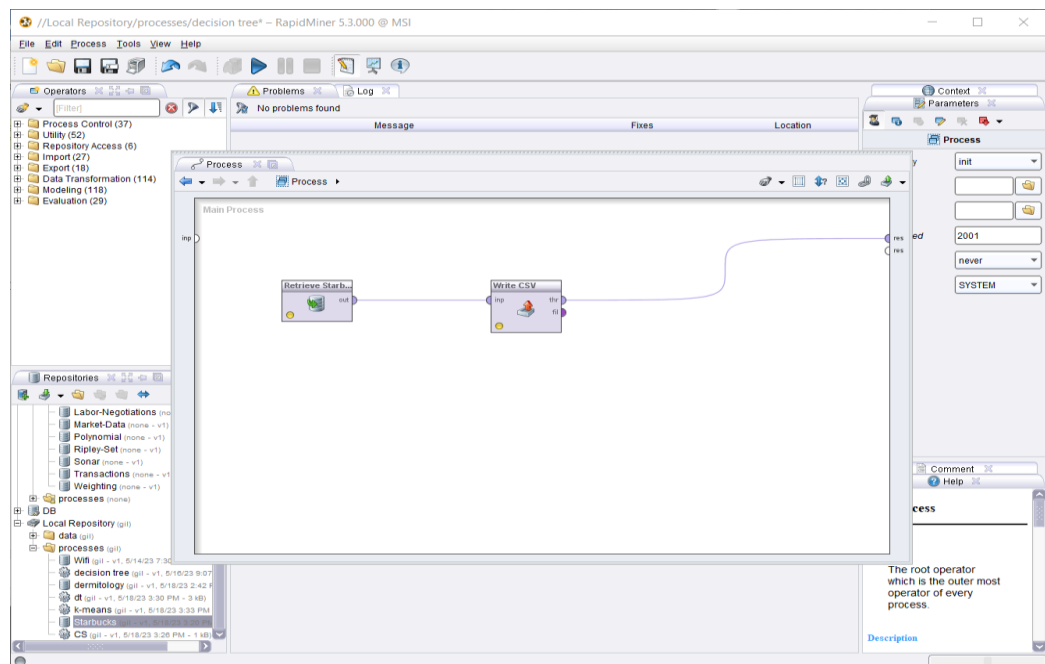
- b) Lalu, klik operator clustering dan ubah nilai k pada parameter menjadi 3 seperti berikut.



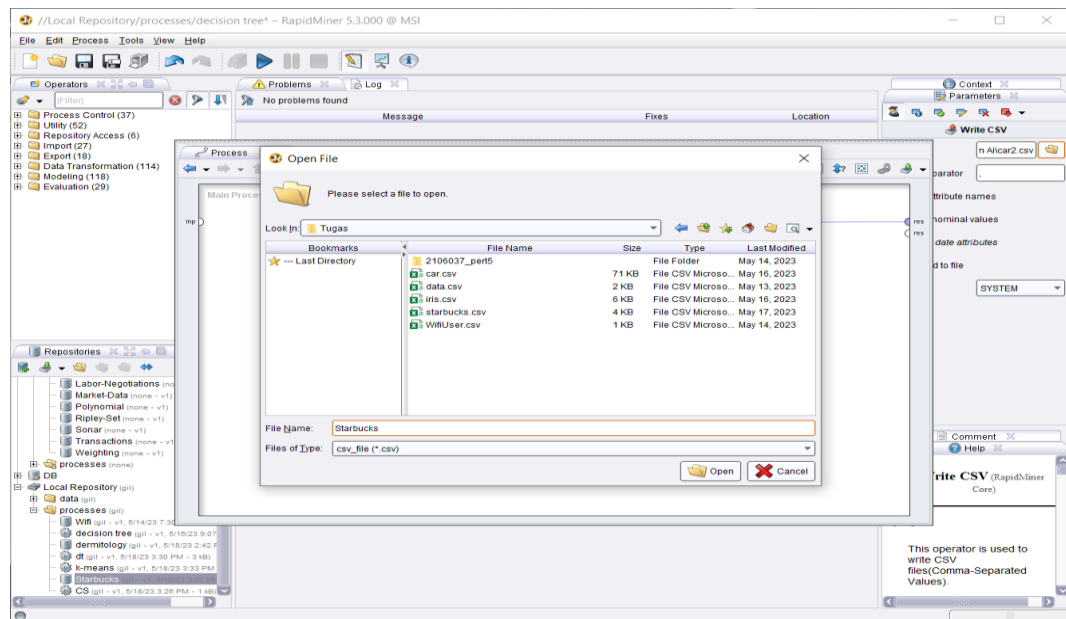
c) Setelah itu, jalankan dan akan tampil seperti berikut.



12) Selanjutnya bisa di delete terlebih dahulu penghubung antara car-out dengan res, dan memasukan operator write CSV, juga hubungkan out-car dengan inp-writeCSV, thr-writeCSV dihubungkan dengan res.



- 13) Kemudian kita bisa klik pada write CSV untuk menampilkan parameter, dan mengubah isi column separator menjadi ',' lalu bisa mengatur penyimpanannya dimana. Untuk terakhirnya bisa langsung coba untuk di run.



2. Tahapan Machine Learning Menggunakan Google Collab(Import Dataset)

- 1) Sebelum mengimport dataset, siapkan terlebih dahulu library yang akan digunakan, berikut merupakan library yang akan digunakan. Kemudian import dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, davies_bouldin_score

dataset = pd.read_csv('Starbukcs.csv')
print('sample data:')
dataset.head()
```

sample data:

	calories	fat	carb	fiber	protein	item
0	350.0	8.0	67.0	5.0	10.0	8-Grain Roll
1	350.0	9.0	64.0	7.0	6.0	Apple Bran Muffin
2	420.0	20.0	59.0	0.0	5.0	Apple Fritter
3	490.0	19.0	75.0	4.0	7.0	Banana Nut Loaf
4	130.0	6.0	17.0	0.0	0.0	Birthday Cake Mini Doughnut

- 2) Cek missing value dan tipe data untuk melihat apakah ada data yang tidak sesuai

```
dataset.isna().sum()

calories    0
fat          0
carb         0
fiber        0
protein      0
item         0
dtype: int64
```

```
dataset.dtypes

calories    float64
fat          float64
carb         float64
fiber        float64
protein      float64
item         object
dtype: object
```

- 3) Menentukan label dan atribut serta membagi dataset menjadi dataset training dan dataset testing

```
x = dataset.iloc[:, :4]
y = dataset['item']

xTrain, xTest, yTrain, yTest = train_test_split(x, y, test_size=0.3, random_state=0)
```

- 4) Decision Tree (Classification)

- a) Melatih model

```
dt = DecisionTreeClassifier()
dt.fit(xTrain, yTrain)
```

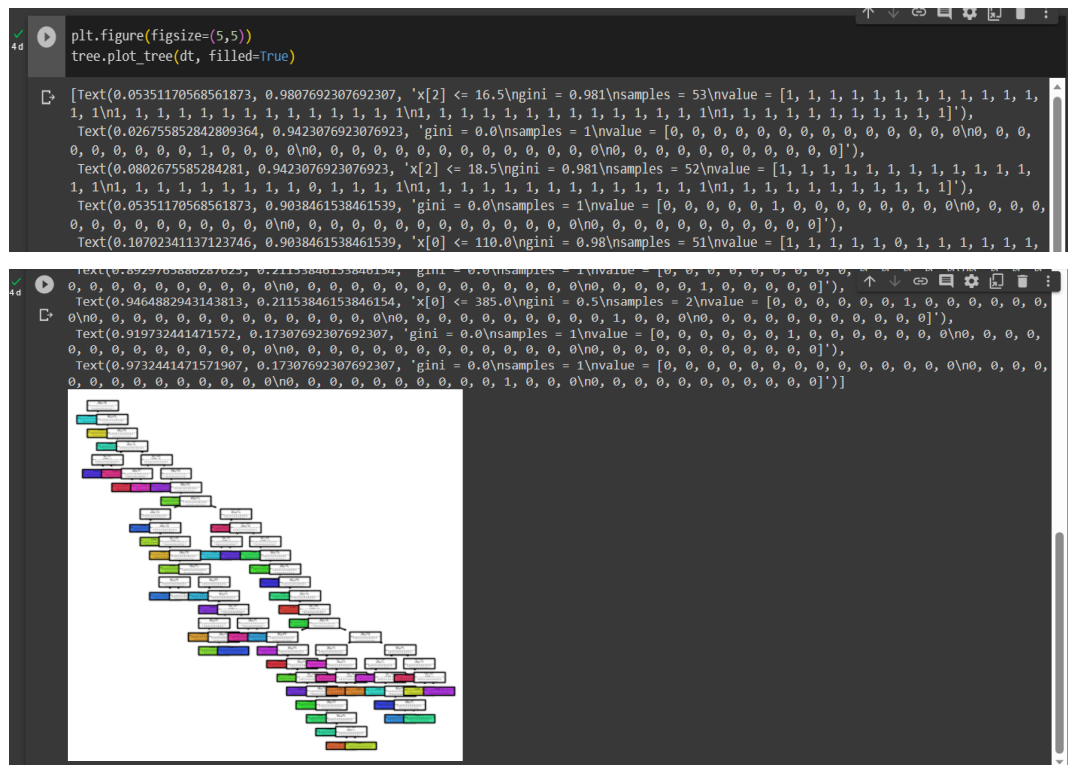
▼ DecisionTreeClassifier
DecisionTreeClassifier()

- b) Nilai akurasi dari model yang sudah dilatih

```
print('Decision Tree Accuracy: {:.3f}'.format(accuracy_score(yTest, dt.predict(xTest))))

Decision Tree Accuracy: 0.000
```

c) Visualisasi model yang sudah dilatih



5) Berikut adalah kode program untuk membuat model menggunakan algoritma K-Means

a) Melatih mode

```
kMeans = KMeans(n_clusters=3)
labels = kMeans.fit_predict(x)
```

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 100 in version 1.2. To silence this warning, you can set `n_init=10` or `n_init='auto'`.

b) Nilai Davies Bouldin Index (DBI) dari model yang sudah dilatih

```
db_score = davies_bouldin_score(x, labels)
print(db_score)
```

0.5255795075161578

c) Visualisasi model yang sudah dilatih

