APPENDIX

CODING PREPROCESISNG DATA BASKET

```
import numpy as np
import pandas as pd
from sklearn import linear_model
import missingno as mno

DataFile=pd.read_csv
   (r"C:/Users/acer/Desktop/CodeDataMin/dataOlimpiade/athlete_even
   ts.csv")

DataOL = DataFile.loc[(DataFile['Sport'].isin(['Basketball']))]
DataFilter = DataOL[['Height', 'Medal']]

isikosong = DataFilter.fillna(0)

isikosong.loc[isikosong.Medal == 'Gold', 'Medal'] = '3'
isikosong.loc[isikosong.Medal == 'Silver', 'Medal'] = '2'
isikosong.loc[isikosong.Medal == 'Bronze', 'Medal'] = '1'
isikosong.loc[isikosong.Height == 0.0, 'Height'] = np.NAN

missing_columns = ['Height']
def random_imputation(df, feature):


    number_missing = isikosong[feature].isnull().sum()
          observed_values   =   isikosong.loc[df[feature].notnull(),
   feature]
    isikosong.loc[isikosong[feature].isnull(), feature + '_imp'] =
   np.random.choice(observed_values,   number_missing,   replace   =
   True)


    return isikosong
for feature in missing_columns:
    isikosong[feature +'_imp'] = isikosong[feature]
    isikosong = random_imputation(isikosong, feature)

deter_data  =  pd.DataFrame(columns  =  ['Det'  +  name  for  name  in
   missing_columns])

for feature in missing_columns:
    deter_data['Det' + feature] = isikosong[feature + '_imp']
                parameters    =    list(set(isikosong.columns)    -
   set(missing_columns) - {feature + '_imp'})


    model = linear_model.LinearRegression()
     model.fit(X = isikosong[parameters], y = isikosong[feature +
   '_imp'])
```

```
    deter_data.loc[isikosong[feature].isnull(), 'Det' + feature] =
   model.predict(isikosong[parameters])
   [isikosong[feature].isnull()]
    mno.matrix(deter_data, figsize = (20, 10))

isikosong.to_csv('C:/Users/acer/Desktop/CodeDataMin/
   dataOlimpiade/DataBasket.csv', index = False)
```

PROSES CLUSTERING DATA BASKET

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import Birch
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from collections import Counter
from sklearn import metrics

df=pd.read_csv('C:/Users/acer/Desktop/CodeDataMin/dataOlimpiade/
   DataBasket.csv')
drop_features = ['Height']
df = df.drop(drop_features, axis = 1)

for k in range(2, 19+1):
    model = Birch(branching_factor = 50, n_clusters = k, threshold
   = 0.1)
    model.fit(df)
    pred = model.predict(df)
    score = silhouette_score(df, pred)
    print('Silhouette Score for k = {}: {:<.3f}'.format(k, score))

model = Birch(branching_factor = 50, n_clusters = 2, threshold =
   0.1)
fit = model.fit(df)
pred = model.predict(df)
labels = model.labels_

cluster = Counter(pred)
cluster

plt.figure(figsize=(10,10))
scatter = plt.scatter(df.iloc[:, 0], df.iloc[:, 1], c= pred)
plt.title("Olimpiade Basket", loc = 'left')
plt.xlabel("Medali")
plt.ylabel("TinggiBadan")
plt.legend(*scatter.legend_elements())
plt.show()



print('Homogeneity :', metrics.homogeneity_score(df['Height_imp'],
   pred))
```

```
print('Completeness:',metrics.completeness_score(df['Height_imp'],
  pred))
print('V-Measure:',metrics.v_measure_score(df['Height_imp'],pred))
```

UNIVERSITAS KATOLIK SOEGIJAPRANATA

CODING PREPROCESISNG DATA BOLA

```
import numpy as np
import pandas as pd
from sklearn import linear_model
import missingno as mno

DataFile=pd.read_csv
   (r"C:/Users/acer/Desktop/CodeDataMin/dataOlimpiade/athlete_even
   ts.csv")

DataOL  =  DataFile.loc[(DataFile['Sport'].isin(['Football','Beach
   Volleyball','Volleyball']))]
DataFilter = DataOL[['Height', 'Medal']]

isikosong = DataFilter.fillna(0)

isikosong.loc[isikosong.Medal == 'Gold', 'Medal'] = '3'
isikosong.loc[isikosong.Medal == 'Silver', 'Medal'] = '2'
isikosong.loc[isikosong.Medal == 'Bronze', 'Medal'] = '1'
isikosong.loc[isikosong.Height == 0.0, 'Height'] = np.NAN

missing_columns = ['Height']
def random_imputation(df, feature):


    number_missing = isikosong[feature].isnull().sum()
          observed_values   =   isikosong.loc[df[feature].notnull(),
   feature]
    isikosong.loc[isikosong[feature].isnull(), feature + '_imp'] =
   np.random.choice(observed_values,   number_missing,   replace   =
   True)


    return isikosong
for feature in missing_columns:
    isikosong[feature +'_imp'] = isikosong[feature]
    isikosong = random_imputation(isikosong, feature)

deter_data  =  pd.DataFrame(columns  =  ['Det'  +  name  for  name  in
   missing_columns])

for feature in missing_columns:
    deter_data['Det' + feature] = isikosong[feature + '_imp']
                parameters    =    list(set(isikosong.columns)    -
   set(missing_columns) - {feature + '_imp'})


    model = linear_model.LinearRegression()
     model.fit(X = isikosong[parameters], y = isikosong[feature +
   '_imp'])
```

```
    deter_data.loc[isikosong[feature].isnull(), 'Det' + feature] =
   model.predict(isikosong[parameters])
   [isikosong[feature].isnull()]
    mno.matrix(deter_data, figsize = (20, 10))

isikosong.to_csv('C:/Users/acer/Desktop/CodeDataMin/
   dataOlimpiade/DataBola.csv', index = False)
```

PROSES CLUSTERING DATA BOLA

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import Birch
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from collections import Counter
from sklearn import metrics

df=pd.read_csv('C:/Users/acer/Desktop/CodeDataMin/dataOlimpiade/
   DataBola.csv')
drop_features = ['Height']
df = df.drop(drop_features, axis = 1)

for k in range(2, 19+1):
    model = Birch(branching_factor = 50, n_clusters = k, threshold
   = 0.1)
    model.fit(df)
    pred = model.predict(df)
    score = silhouette_score(df, pred)
    print('Silhouette Score for k = {}: {:<.3f}'.format(k, score))

model = Birch(branching_factor = 50, n_clusters = 2, threshold =
   0.1)
fit = model.fit(df)
pred = model.predict(df)
labels = model.labels_

cluster = Counter(pred)
cluster

plt.figure(figsize=(10,10))
scatter = plt.scatter(df.iloc[:, 0], df.iloc[:, 1], c= pred)
plt.title("Olimpiade Basket", loc = 'left')
plt.xlabel("Medali")
plt.ylabel("TinggiBadan")
plt.legend(*scatter.legend_elements())
plt.show()


print('Homogeneity :', metrics.homogeneity_score(df['Height_imp'],
   pred))
```

```
print('Completeness:',metrics.completeness_score(df['Height_imp'],
  pred))
print('V-Measure:',metrics.v_measure_score(df['Height_imp'],pred))
```

CODING PREPROCESISNG DATA SEMUA

```
import numpy as np
import pandas as pd
from sklearn import linear_model
import missingno as mno

DataFile=pd.read_csv
   (r"C:/Users/acer/Desktop/CodeDataMin/dataOlimpiade/athlete_even
  ts.csv")

DataFilter = DataOL[['Height', 'Medal']]

isikosong = DataFilter.fillna(0)

isikosong.loc[isikosong.Medal == 'Gold', 'Medal'] = '3'
isikosong.loc[isikosong.Medal == 'Silver', 'Medal'] = '2'
isikosong.loc[isikosong.Medal == 'Bronze', 'Medal'] = '1'
isikosong.loc[isikosong.Height == 0.0, 'Height'] = np.NAN

missing_columns = ['Height']
def random_imputation(df, feature):

    number_missing = isikosong[feature].isnull().sum()
          observed_values   =   isikosong.loc[df[feature].notnull(),
  feature]
    isikosong.loc[isikosong[feature].isnull(), feature + '_imp'] =
  np.random.choice(observed_values,   number_missing,   replace   =
  True)

    return isikosong
for feature in missing_columns:
    isikosong[feature +'_imp'] = isikosong[feature]
    isikosong = random_imputation(isikosong, feature)

deter_data  =  pd.DataFrame(columns  =  ['Det'  +  name  for  name  in
  missing_columns])

for feature in missing_columns:
    deter_data['Det' + feature] = isikosong[feature + '_imp']
                parameters    =    list(set(isikosong.columns)    -
  set(missing_columns) - {feature + '_imp'})

    model = linear_model.LinearRegression()
     model.fit(X = isikosong[parameters], y = isikosong[feature +
  '_imp'])

    deter_data.loc[isikosong[feature].isnull(), 'Det' + feature] =
  model.predict(isikosong[parameters])
  [isikosong[feature].isnull()]
    mno.matrix(deter_data, figsize = (20, 10))
```

```
isikosong.to_csv('C:/Users/acer/Desktop/CodeDataMin/
  dataOlimpiade/DataSemua.csv', index = False)
```

PROSES CLUSTERING DATA SEMUA

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import Birch
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from collections import Counter
from sklearn import metrics

df=pd.read_csv('C:/Users/acer/Desktop/CodeDataMin/dataOlimpiade/
  DataSemua.csv')
drop_features = ['Height']
df = df.drop(drop_features, axis = 1)

for k in range(2, 19+1):
    model = Birch(branching_factor = 50, n_clusters = k, threshold
  = 0.1)
    model.fit(df)
    pred = model.predict(df)
    score = silhouette_score(df, pred)
    print('Silhouette Score for k = {}: {:<.3f}'.format(k, score))

model = Birch(branching_factor = 50, n_clusters = 2, threshold =
  0.1)
fit = model.fit(df)
pred = model.predict(df)
labels = model.labels_

cluster = Counter(pred)
cluster

plt.figure(figsize=(10,10))
scatter = plt.scatter(df.iloc[:, 0], df.iloc[:, 1], c= pred)
plt.title("Olimpiade Basket", loc = 'left')
plt.xlabel("Medali")
plt.ylabel("TinggiBadan")
plt.legend(*scatter.legend_elements())
plt.show()

print('Homogeneity :', metrics.homogeneity_score(df['Height_imp'],
  pred))
print('Completeness:',metrics.completeness_score(df['Height_imp'],
  pred))
print('V-Measure:',metrics.v_measure_score(df['Height_imp'],pred))
```

PLAGIARISM
CHECK.ORG

4.39% PLAGIARISM APPROXIMATELY

Report #14318943

CHAPTER 1 Introduction Background The Olympics is an international sporting event that is held once every four years and has been around for 120 years, which includes summer and winter sports. attended by more than 200 countries and tens of thousands of athletes. Various kinds of athletes take part in the Olympics to win medals, there are gold, silver, and bronze medals. From old to young, men and women, lightweight and heavyweight, tall people and short people. The general opinion of people at the Olympics, in general, is that if short people take part in basketball competitions, they will not win medals. From there the problem can be found, the public stigma of short athletes if they participate in the basketball olympics will definitely lose or have trouble when competing, because the average height of basketball people is very high. This study will prove whether the stigma of society is true if the height of a basketball athlete can determine whether an athlete wins a medal or not. The data used in the form