

**PROJECT**  
**UJIAN AKHIR SEMESTER**  
**KONSEP DAN APLIKASI DATA MINING**



Oleh :

Diahawa Fitaloka (17.51.0002)

**PROGRAM STUDI S1- SISTEM INFORMASI**

**KEMENTRIAN RISET DAN TEKNOLOGI PENDIDIKAN TINGGI**  
**SEKOLAH TINGGI MANAJEMEN INFORMATIKA KOMPUTER**  
**PRADNYA PARAMITA**  
**MALANG**  
**2020**

## 1. Soal No 1.

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import csv
```

```
In [3]: data =pd.read_csv('E:/vita/UASdm/uasdmno1/dataset_soalNo.1.csv',delimiter=";")
```

```
In [4]: data.head()
```

```
Out[4]:
```

	Age	Income	Student	Credit_rating	Class (buy_computer)
0	<=30	High	No	Fair	No
1	<=30	High	No	Excellent	No
2	31..40	High	No	Fair	Yes
3	> 40	Medium	No	Fair	Yes
4	> 40	Low	Yes	Fair	Yes

```
In [5]: data.tail(10)
```

```
Out[5]:
```

	Age	Income	Student	Credit_rating	Class (buy_computer)
41	> 40	Low	Yes	Fair	No
42	31..40	Low	Yes	Fair	Yes
43	31..40	Low	Yes	Excellent	No
44	<= 30	High	No	Excellent	No
45	<= 30	Medium	Yes	Excellent	Yes
46	> 40	Low	Yes	Fair	Yes
47	<= 30	Low	Yes	Fair	Yes
48	31..40	Medium	No	Fair	No
49	31..40	High	Yes	Excellent	Yes
50	> 40	Medium	No	Excellent	No

```
In [6]: data['Age'].value_counts()
```

```
Out[6]: > 40      17  
        <= 30     15  
        31..40    14  
        <=30      5  
        Name: Age, dtype: int64
```

```
In [7]: data['Income'].value_counts()
```

```
Out[7]: Low      21  
        Medium   19  
        High     11  
        Name: Income, dtype: int64
```

```
In [8]: data['Student'].value_counts()
```

```
Out[8]: Yes      27  
        No       24  
        Name: Student, dtype: int64
```

```
In [9]: data['Credit_rating'].value_counts()
```

```
Out[9]: Fair      31  
        Excellent  20  
        Name: Credit_rating, dtype: int64
```

```
In [10]: data['Class (buy_computer)'].value_counts()
```

```
Out[10]: Yes      29  
        No       22  
        Name: Class (buy_computer), dtype: int64
```

```
In [11]: data.shape
```

```
Out[11]: (51, 5)
```

```
In [12]: PYes = 27/51  
        PNo = 24/51
```

```
In [13]: pd.crosstab(data['Age'], data['Income'])
```

Out[13]:

Income	High	Low	Medium
Age			
31..40	5	5	4
<= 30	1	6	8
<=30	5	0	0
> 40	0	10	7

In [14]: pd.crosstab(data['Age'],data['Student'])

Out[14]:

Student	No	Yes
Age		
31..40	7	7
<= 30	6	9
<=30	5	0
> 40	6	11

In [15]: pd.crosstab(data['Age'],data['Credit\_rating'])

Out[15]:

Credit_rating	Excellent	Fair
Age		
31..40	7	7
<= 30	4	11
<=30	3	2
> 40	6	11

In [16]: pd.crosstab(data['Age'],data['Class (buy\_computer)'])

Out[16]:

Class (buy_computer)	No	Yes
Age		
31..40	4	10
<= 30	7	8
<=30	3	2
> 40	8	9

In [17]: pd.crosstab(data['Age'],data['Income'])

Out[17]:

Income	High	Low	Medium
Age			
31..40	5	5	4
<= 30	1	6	8
<=30	5	0	0
> 40	0	10	7

In [19]: pd.crosstab(data['Age'],data['Income'])

Out[19]:

Income	High	Low	Medium
Age			
31..40	5	5	4
<= 30	1	6	8
<=30	5	0	0
> 40	0	10	7

In [20]: pd.crosstab(data['Age'],data['Class (buy\_computer)'])

Out[20]:

Class (buy_computer)	No	Yes
Age		
31..40	4	10
<= 30	7	8
<=30	3	2
> 40	8	9

In [21]: `pd.crosstab(data['Income'],data['Class (buy_computer)'])`

Out[21]:

Class (buy_computer)	No	Yes
Income		
High	6	5
Low	11	10
Medium	5	14

In [22]: `pd.crosstab(data['Income'],data['Credit_rating'])`

Out[22]:

Credit_rating	Excellent	Fair
Income		
High	5	6
Low	8	13
Medium	7	12

In [23]: `pd.crosstab(data['Income'],data['Age'])`

Out[23]:

Age	31..40	<= 30	<=30	> 40
Income				
High	5	1	5	0
Low	5	6	0	10
Medium	4	8	0	7

```
In [24]: PHighNo = 6/22  
        PLowNo = 11/22  
        PMediumNo = 5/22  
  
        PHighYes = 5/29  
        PLowYes = 10/29  
        PMediumYes = 5/29  
  
        PHigh = 11/52  
        PLow = 21/51  
        PMedium = 19/51
```

```
print (PHighNo)
```

```
0.2727272727272727
```

```
In [25]: print (PHighYes)
```

```
0.1724137931034483
```

```
In [26]: print (PHigh)
```

```
0.21153846153846154
```

```
In [27]: print (PLowNo)
```

```
0.5
```

```
In [28]: print (PLowYes)
```

```
0.3448275862068966
```

```
In [29]: print (PLow)
```

```
0.4117647058823529
```

```
In [30]: print (PMediumYes)
```

```
0.1724137931034483
```

```
In [31]: print (PMediumNo)
```

```
0.22727272727272727
```

```
In [32]: print (PMedium)
```

```
0.37254901960784315
```

## 2. Soal No 2

- a. Apabila Cuaca buruk dengan nilai = 1, Weekday, dan Game = 0, maka berapa roti yang harus dibuat?

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: data = pd.read_excel('E:/vita/UASdm/uasdmno2/dataset_soal No2.xls')
```

```
In [3]: data
```

```
Out[3]:
```

	Category	WeatherV-1	HolidayV-2	GameV-3	Qty
0	A	5	1	0	250
1	B	3	1	1	200
2	C	1	1	0	75
3	D	4	1	1	400
4	E	4	0	0	150
5	F	2	0	0	50

```
In [4]: import math
dis = []
for i in range(6):
    dis.append(math.sqrt((float(data.iloc[i]['WeatherV-1'])-1)**2+(float(data.iloc[i]['HolidayV-2'])-1)**2+(float(data.iloc[i]['GameV-3'])-0)**2))
```

```
In [5]: data['dis'] = dis
data
```

```
Out[5]:
```

	Category	WeatherV-1	HolidayV-2	GameV-3	Qty	dis
0	A	5	1	0	250	4.000000
1	B	3	1	1	200	2.236068
2	C	1	1	0	75	0.000000
3	D	4	1	1	400	3.162278
4	E	4	0	0	150	3.162278
5	F	2	0	0	50	1.414214

- b. Apabila Cuaca baik dengan nilai 4, Weekend, dan Game =1, maka berapa roti yang harus dibuat?

```
In [14]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [15]: data = pd.read_excel('E:/vita/UASdm/dataset_soal No. 2.xls')
```

```
In [16]: data
```

```
Out[16]:
```

	Category	WeatherV-1	HolidayV-2	GameV-3	Qty
0	A	5	1	0	250
1	B	3	1	1	200
2	C	1	1	0	75
3	D	4	1	1	400
4	E	4	0	0	150
5	F	2	0	0	50

```
In [17]: import math
dis = []
for i in range(6):
    dis.append(math.sqrt((float(data.iloc[i]['WeatherV-1'])-4)**2+(float(data.iloc[i]['HolidayV-2'])-1)**2+(float(data.iloc[i]['GameV-3'])-1)**2))
```

```
In [18]: data['dis'] = dis
data
```

```
Out[18]:
```

	Category	WeatherV-1	HolidayV-2	GameV-3	Qty	dis
0	A	5	1	0	250	1.414214
1	B	3	1	1	200	1.000000
2	C	1	1	0	75	3.162278
3	D	4	1	1	400	0.000000
4	E	4	0	0	150	1.414214
5	F	2	0	0	50	2.449490



### 3. Soal No 3

```
In [1]: import numpy as np
import pandas as pd
from apyori import apriori
```

```
In [15]: store_data = pd.read_excel ('E:vita/UASdm/soaldmno3/dataset_soalNo.3.xlsx')
```

```
In [16]: store_data.head()
```

```
Out[16]:
```

	Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8	Item9	Item10
0	burgers	meatballs	eggs	low fat yogurt	NaN	mineral water	salmon	low fat yogurt	NaN	mineral water
1	chutney	low fat yogurt	NaN	whole wheat pasta	french fries	mineral water	salmon	whole wheat pasta	french fries	mineral water
2	turkey	whole wheat pasta	french fries	soup	light cream	shallot	NaN	soup	light cream	shallot
3	mineral water	soup	light cream	frozen vegetables	spaghetti	green tea	NaN	frozen vegetables	spaghetti	green tea
4	low fat yogurt	frozen vegetables	spaghetti	french fries	eggs	chocolate	frozen smoothie	french fries	eggs	chocolate

```
In [17]: store_data.tail()
```

```
In [6]: store_data.tail()
```

```
Out[6]:
```

	Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8	Item9	Item10
2049	burgers	eggs	french fries	fresh tuna	spaghetti	olive oil	clothes accessories	turkey	eggs	french fries
2050	burgers	eggs	frozen smoothie	french wine	eggs	french fries	energy drink	french fries	NaN	chocolate
2051	whole wheat pasta	cake	melons	champagne	pancakes	light mayo	soup	chocolate	milk	herb & pepper
2052	ground beef	tomato sauce	spaghetti	red wine	honey	hot dogs	turkey	herb & pepper	whole wheat pasta	mineral water
2053	burgers	eggs	frozen smoothie	milk	bacon	eggs	french fries	mineral water	avocado	cookies

```
In [7]: store_data.shape
```

```
Out[7]: (2054, 10)
```

```
In [8]: records = []
for i in range (0, 2054):
    records.append ([str(store_data.values[i,j])for j in range (0, 10)])
```

```
In [9]: association_rules = apriori (records, min_support=0.2,min_confidence=0.2,min_lenght=2)
association_results = list (association_rules)
```

```
In [10]: print(len(association_results))
```

61

```
In [11]: print (association_results[0])
```

```
RelationRecord(items=frozenset({'avocado'}), support=0.314508276533593, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'avocado'}), confidence=0.314508276533593, lift=1.0)])
```

```
In [12]: results = []
for item in association_results:
    pair = item[0]
    items = [X for X in pair]

    value0 = str(items[0])
    value1 = str(item[1])
    value2 = str(item[1])[:10]
    value3 = str(item[2][0][2])[:10]
    value4 = str(item[2][0][3])[:10]

    rows = (value0,value1,value2,value3,value4)

    results.append(rows)

label = ['title1', ['title2'], ['support'], ['confidence'], ['lift']]

store_suggestion = pd.DataFrame.from_records(results,columns=label)

print (store_suggestion)
```

```
(title1,) (title2,) (support,) (confidence,) (lift,)
0 avocado 0.314508276533593 0.31450827 0.31450827 1.0
(title1,) (title2,) (support,) (confidence,) (lift,)
0 avocado 0.314508276533593 0.31450827 0.31450827 1.0
1 burgers 0.24294060370009737 0.24294060 0.24294060 1.0
(title1,) (title2,) (support,) (confidence,) (lift,)
0 avocado 0.314508276533593 0.31450827 0.31450827 1.0
1 burgers 0.24294060370009737 0.24294060 0.24294060 1.0
2 chocolate 0.4756572541382668 0.47565725 0.47565725 1.0
(title1,) (title2,) (support,) (confidence,) (lift,)
0 avocado 0.314508276533593 0.31450827 0.31450827 1.0
1 burgers 0.24294060370009737 0.24294060 0.24294060 1.0
2 chocolate 0.4756572541382668 0.47565725 0.47565725 1.0
3 clothes accessories 0.33982473222979553 0.33982473 0.33982473 1.0
(title1,) (title2,) (support,) (confidence,) (lift,)
0 avocado 0.314508276533593 0.31450827 0.31450827 1.0
1 burgers 0.24294060370009737 0.24294060 0.24294060 1.0
2 chocolate 0.4756572541382668 0.47565725 0.47565725 1.0
3 clothes accessories 0.33982473222979553 0.33982473 0.33982473 1.0
```

55	french fries	0.2911392405063291	0.29113924	0.29113924	1.0
56	french fries	0.20837390457643623	0.20837390	0.20837390	1.0
57	french fries	0.2249269717624148	0.22492697	0.22492697	1.0
58	french fries	0.22249269717624148	0.22249269	0.22249269	1.0
59	energy drink	0.23661148977604674	0.23661148	0.23661148	1.0
	(title1,)	(title2,)	(support,)	(confidence,)	(lift,)
0	avocado	0.314508276533593	0.31450827	0.31450827	1.0
1	burgers	0.24294060370009737	0.24294060	0.24294060	1.0
2	chocolate	0.4756572541382668	0.47565725	0.47565725	1.0
3	clothes accessories	0.33982473222979553	0.33982473	0.33982473	1.0
4	cookies	0.3588120740019474	0.35881207	0.35881207	1.0
..	...	...	...	...	...
56	french fries	0.20837390457643623	0.20837390	0.20837390	1.0
57	french fries	0.2249269717624148	0.22492697	0.22492697	1.0
58	french fries	0.22249269717624148	0.22249269	0.22249269	1.0
59	energy drink	0.23661148977604674	0.23661148	0.23661148	1.0
60	eggs	0.24196689386562803	0.24196689	0.24196689	1.0
[61 rows x 5 columns]					

In [13]: `store_suggestion.describe()`

Out[13]:

	(title1,)	(title2,)	(support,)	(confidence,)	(lift,)
count	61	61	61	61	61
unique	15	53	53	53	1
top	french fries	0.24294060370009737	0.24294060	0.24294060	1.0
freq	18	4	4	4	61

In [14]: `store_suggestion.to_excel('E:/vita/UASdm/outputsoalno3.xls')`

In [ ]:

#### 4. Soal No 4

- a. Berdasarkan data tersebut bagaimana perlakuan dengan kondisi Ibu hamil dengan Usia 30 Tahun, yang merupakan Kelahiran ke -1, dengan Waktu kelahiran sesuai dengan HPL, Memiliki tekanan darah Normal? Carilah KNN dengan menggunakan Key = 5

```
In [5]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [6]: pd.__version__
```

```
Out[6]: '1.0.1'
```

```
In [7]: data = pd.read_csv('E:/vita/UASdm/uasdmno4/dataset_soal No. 4.txt',
delimiter=',')
```

```
In [8]: data
```

```
Out[8]:
```

	Usia	Kelahiran_ke-	Waktu_Kelahiran	Tekanan_darah	Kelainan_jantung	Caesarian
0	22	1	0	2	0	0
1	26	2	0	1	0	1
2	26	2	1	1	0	0
3	28	1	0	2	0	0
4	22	2	0	1	0	1
...	...	...	...	...	...	...
75	27	2	1	1	0	0
76	33	4	0	1	0	1
77	29	2	1	2	0	1
78	25	1	2	0	0	1
79	24	2	2	1	0	0

80 rows x 6 columns

```
In [9]: import math
dis = []
for i in range(80):
    dis.append(math.sqrt((float(data.iloc[i]['Usia'])-30)**2+
(float(data.iloc[i]['Kelahiran_ke-']- 1)**2+
(float(data.iloc[i]['Waktu_Kelahiran'])-0)**2+
(float(data.iloc[i]['Tekanan_darah'])-1)**2))
```

```
In [10]: data['dis'] = dis
data
```

Out[10]:

	Usia	Kelahiran_ke-	Waktu_Kelahiran	Tekanan_darah	Kelainan_jantung	Caesarian	dis
0	22	1	0	2	0	0	8.062258
1	26	2	0	1	0	1	4.123106
2	26	2	1	1	0	0	4.242641
3	28	1	0	2	0	0	2.236068
4	22	2	0	1	0	1	8.062258
...	...	...	...	...	...	...	...
75	27	2	1	1	0	0	3.316625
76	33	4	0	1	0	1	4.242641
77	29	2	1	2	0	1	2.000000
78	25	1	2	0	0	1	5.477226
79	24	2	2	1	0	0	6.403124

80 rows × 7 columns

In [11]: data.sort\_values('dis')

Out[11]:

	Usia	Kelahiran_ke-	Waktu_Kelahiran	Tekanan_darah	Kelainan_jantung	Caesarian	dis
27	30	1	0	1	0	0	0.000000
38	31	1	0	1	0	0	1.000000
67	29	2	0	1	1	0	1.414214
54	29	2	0	1	1	1	1.414214
59	30	2	1	2	1	1	1.732051
...	...	...	...	...	...	...	...
41	19	1	0	1	0	1	11.000000
61	19	1	0	1	0	1	11.000000
25	18	1	0	1	0	0	12.000000
26	18	1	1	2	1	1	12.083046
70	17	1	0	0	0	1	13.038405

80 rows × 7 columns

In [12]: y = data.sort\_values('dis').head(5)  
y

Out[12]:

	Usia	Kelahiran_ke-	Waktu_Kelahiran	Tekanan_darah	Kelainan_jantung	Caesarian	dis
27	30	1	0	1	0	0	0.000000
38	31	1	0	1	0	0	1.000000
67	29	2	0	1	1	0	1.414214
54	29	2	0	1	1	1	1.414214
59	30	2	1	2	1	1	1.732051

In [13]: z = y["Caesarian"]  
z

Out[13]:

```

27    0
38    0
67    0
54    1
59    1
Name: Caesarian, dtype: int64

```

In [14]: np.mean(z)

Out[14]: 0.4

In [15]: data.to\_excel('E:/vita/UASdm/uasdmno4/keluaran\_Nomer4a.xls')

In [ ]:

- b. Bagaimana Apabila Ibu hamil dengan Usia 29 Tahun, yang merupakan Kelahiran ke -2, dengan Waktu kelahiran sesuai dengan HPL, Memiliki tekanan darah Tinggi? Carilah KNN dengan menggunakan Key = 5

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: pd.__version__
```

```
Out[2]: '1.0.1'
```

```
In [3]: data = pd.read_csv('E:/vita/UASdm/uasdmno4/dataset_soal No. 4.txt', delimiter=',')
```

```
In [4]: data
```

```
Out[4]:
```

	Usia	Kelahiran_ke-	Waktu_Kelahiran	Tekanan_darah	Kelainan_jantung	Caesarian
0	22	1	0	2	0	0
1	26	2	0	1	0	1
2	26	2	1	1	0	0
3	28	1	0	2	0	0
4	22	2	0	1	0	1
...	...	...	...	...	...	...
75	27	2	1	1	0	0
76	33	4	0	1	0	1
77	29	2	1	2	0	1
78	25	1	2	0	0	1
79	24	2	2	1	0	0

80 rows x 6 columns

```
In [5]: import math
dis = []
for i in range(80):
    dis.append(math.sqrt((float(data.iloc[i]['Usia'])-29)**2+
                        (float(data.iloc[i]['Kelahiran_ke-'])- 2)**2+
                        (float(data.iloc[i]['Waktu_Kelahiran'])-0)**2+
                        (float(data.iloc[i]['Tekanan_darah'])-2)**2))
```

```
In [6]: data['dis'] = dis
data
```

Out[6]:

	Usia	Kelahiran_ke-	Waktu_Kelahiran	Tekanan_darah	Kelainan_jantung	Caesarian	dis
0	22	1	0	2	0	0	7.071068
1	26	2	0	1	0	1	3.162278
2	26	2	1	1	0	0	3.316625
3	28	1	0	2	0	0	1.414214
4	22	2	0	1	0	1	7.071068
...	...	...	...	...	...	...	...
75	27	2	1	1	0	0	2.449490
76	33	4	0	1	0	1	4.582576
77	29	2	1	2	0	1	1.000000
78	25	1	2	0	0	1	5.000000
79	24	2	2	1	0	0	5.477226

80 rows x 7 columns

```
In [7]: data.sort_values('dis')
```

Out[7]:

	Usia	Kelahiran_ke-	Waktu_Kelahiran	Tekanan_darah	Kelainan_jantung	Caesarian	dis
54	29	2	0	1	1	1	1.000000
77	29	2	1	2	0	1	1.000000
67	29	2	0	1	1	0	1.000000
59	30	2	1	2	1	1	1.414214
3	28	1	0	2	0	0	1.414214
...	...	...	...	...	...	...	...
61	19	1	0	1	0	1	10.099505
26	18	1	1	2	1	1	11.090537
31	40	1	0	1	1	1	11.090537

31	40	1	0	1	1	1	11.090537
25	18	1	0	1	0	0	11.090537
70	17	1	0	0	0	1	12.206556

80 rows x 7 columns

```
In [8]: y = data.sort_values('dis').head(5)
y
```

Out[8]:

	Usia	Kelahiran_ke-	Waktu_Kelahiran	Tekanan_darah	Kelainan_jantung	Caesarian	dis
54	29	2	0	1	1	1	1.000000
77	29	2	1	2	0	1	1.000000
67	29	2	0	1	1	0	1.000000
59	30	2	1	2	1	1	1.414214
3	28	1	0	2	0	0	1.414214

```
In [9]: z = y["Caesarian"]
z
```

Out[9]:

54	1
77	1
67	0
59	1
3	0

Name: Caesarian, dtype: int64

```
In [10]: np.mean(z)
```

Out[10]: 0.6

```
In [11]: data.to_excel('E:/vita/UASdm/uasdmno4/keluaran_no4b.xls')
```