Bagian Pertama

In [3]:

```
import pandas as pd

df = pd.DataFrame({
    'country' : ['India','US','Japan','US','Japan'],
    'age' : [44,34,46,35,23],
    'salaty' : [72000,65000,98000,45000,34000]
})
df
```

Out[3]:

	country	age	salaty
0	India	44	72000
1	US	34	65000
2	Japan	46	98000
3	US	35	45000
4	Japan	23	34000

In [4]:

```
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
df['country'] = label_encoder.fit_transform(df['country'])
df
```

Out[4]:

	country	age	salaty
0	0	44	72000
1	2	34	65000
2	1	46	98000
3	2	35	45000
4	1	23	34000

In [5]:

```
label_encoder.classes_
```

Out[5]:

```
array(['India', 'Japan', 'US'], dtype=object)
```

In [6]:

```
df = pd.DataFrame({
    'country' : ['India','US','Japan','US','Japan'],
    'age' : [44,34,46,35,23],
    'salaty' : [72000,65000,98000,45000,34000]
})
df
```

Out[6]:

	country	age	salaty

	country	age	salaty
0	India	44	72000
1	US	34	65000
2	Japan	46	98000
3	US	35	45000
4	Japan	23	34000

In [7]:

```
X = df['country'].values.reshape(-1,1)
X
```

Out[7]:

```
array([['India'],
       ['US'],
       ['Japan'],
       ['US'],
       ['Japan']], dtype=object)
```

In [8]:

```
from sklearn.preprocessing import OneHotEncoder

onehot_encoder = OneHotEncoder()
X = onehot_encoder.fit_transform(X).toarray()
X
```

Out[8]:

```
array([[1., 0., 0.],
       [0., 0., 1.],
       [0., 1., 0.],
       [0., 0., 1.],
       [0., 1., 0.]])
```

In [9]:

```
onehot_encoder.categories_
```

Out[9]:

```
[array(['India', 'Japan', 'US'], dtype=object)]
```

In [10]:

```
df_onehot = pd.DataFrame(X, columns=[str(i) for i in range(X.shape[1])])
df_onehot
```

Out[10]:

	0	1	2
0	1.0	0.0	0.0
1	0.0	0.0	1.0
2	0.0	1.0	0.0
3	0.0	0.0	1.0
4	0.0	1.0	0.0

In [11]:

```
df = pd.concat([df_onehot, df], axis=1)
df
```

Out[11]:

	0	1	2	country	age	salaty

	0	1	2	country	age	salaty
0	1.0	0.0	0.0	India	44	72000
1	0.0	0.0	1.0	US	34	65000
2	0.0	1.0	0.0	Japan	46	98000
3	0.0	0.0	1.0	US	35	45000
4	0.0	1.0	0.0	Japan	23	34000

In [12]:

```
df = df.drop(['country'], axis = 1)
df
```

Out[12]:

	0	1	2	age	salaty
0	1.0	0.0	0.0	44	72000
1	0.0	0.0	1.0	34	65000
2	0.0	1.0	0.0	46	98000
3	0.0	0.0	1.0	35	45000
4	0.0	1.0	0.0	23	34000

Bagian Kedua

In [13]:

```
corpus = [
    'The Industrial Revolution 4.0 is changing most of the business activities',
    'The Industrial Revolution 4.0 has five technologies that are the main points',
    'Industry 4.0 opens new challenges for companies'
]

corpus
```

Out[13]:

```
['The Industrial Revolution 4.0 is changing most of the business activities',
 'The Industrial Revolution 4.0 has five technologies that are the main points',
 'Industry 4.0 opens new challenges for companies']
```

In [14]:

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer()
vectorizer_X = vectorizer.fit_transform(corpus).todense()
vectorizer_X
```

Out[14]:

```
matrix([[1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
         2],
        [0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1,
         2],
        [0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
         0]], dtype=int64)
```

In [15]:

```
vectorizer.get_feature_names()
```

Out[15]:

```
['activities',
 'are',
 'business',
 'challenges',
```

```
 'changing',
 'companies',
 'five',
 'for',
 'has',
 'industrial',
 'industry',
 'is',
 'main',
 'most',
 'new',
 'of',
 'opens',
 'points',
 'revolution',
 'technologies',
 'that',
 'the']
```

In [16]:

```
from sklearn.metrics.pairwise import euclidean_distances

for i in range(len(vectorizer_X)):
    for j in range(i, len(vectorizer_X)):
        if i==j:
            continue
        jarak = euclidean_distances(vectorizer_X[i], vectorizer_X[j])
        print(f'Jarak dokumen {i+1} dan {j+1}: {jarak}')
```

```
Jarak dokumen 1 dan 2: [[3.60555128]]
Jarak dokumen 1 dan 3: [[4.24264069]]
Jarak dokumen 2 dan 3: [[4.35889894]]
```

In [21]:

```
vectorizer = CountVectorizer(stop_words='english')
vectorizer_X = vectorizer.fit_transform(corpus).todense()
vectorizer_X
```

Out[21]:

```
matrix([[1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1],
        [0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0]], dtype=int64)
```

In [22]:

```
vectorizer.get_feature_names()
```

Out[22]:

```
['activities',
 'business',
 'challenges',
 'changing',
 'companies',
 'industrial',
 'industry',
 'main',
 'new',
 'opens',
 'points',
 'revolution',
 'technologies']
```

In [ ]: