**2 Cara Implementasi One-Hot Encoding di Python**

*Encoding* adalah salah satu tahap praproses data sebelum diproses dengan algoritma *machine learning*. Dalam mengerjakan projek *data science* ataupun *machine learning*, kita akan sangat mungkin menemukan satu atau beberapa fitur yang bertipe kategori, misalnya ‘Sangat Baik’, ‘Baik’, ‘Tidak Baik. Nah, komputer tidak dapat memproses data bertipe kategori sehingga kita harus mengubah data tersebut menjadi berbentuk bilangan. Proses ini disebut dengan *encoding*.

**Apa itu One-Hot Encoding dan kenapa perlu dilakukan?**

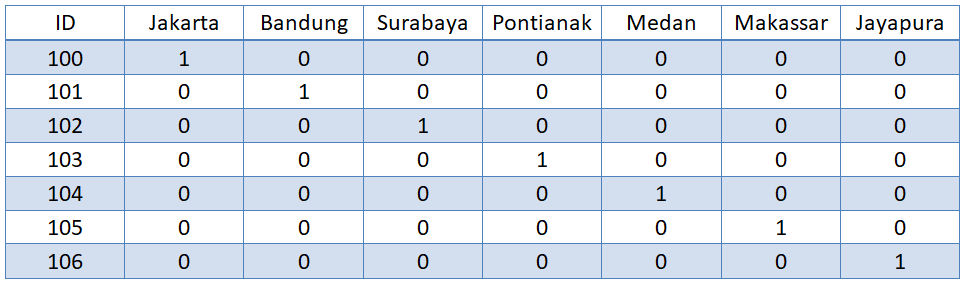
One-Hot encoding adalah salah satu metode *encoding*. Metode ini merepresentasikan data bertipe kategori sebagai vektor biner yang bernilai integer, 0 dan 1, dimana semua elemen akan bernilai 0 kecuali satu elemen yang bernilai 1, yaitu elemen yang memiliki nilai kategori tersebut.

Perhatikan contoh berikut. Misalnya kita memiliki data nama kota beserta ID-nya seperti di bawah ini.



Dalam prakteknya menggunakan Python, banyak algoritma *machine learning* yang tidak dapat memproses data kota tersebut karena bertipe kategori. Untuk memprosesnya, kita dapat mengubah fitur ‘City’ tersebut menjadi bilangan dengan one-hot encoding.

Setiap kategori pada fitur ‘City’ akan menjadi kolom/fitur baru yang bernilai 0 atau 1. Misalnya, pada kolom ‘Jakarta’ pada indeks dengan ID=100 akan bernilai 1, sementara indeks lainnya akan bernilai 0. Nantinya setelah diterapkan one-hot encoding, data di atas kira-kira akan menjadi seperti di bawah ini.



**2 Cara One-Hot Encoding di Python**

Ada dua cara yang dapat Anda lakukan untuk mengimplementasikan one-hot encoding di Python, yaitu dengan menggunakan *library* Pandas dan Scikit-Learn.

**One-Hot Encoding menggunakan Pandas**

Dalam contoh penerapan one-hot encoding di Python ini, kita akan menggunakan data sederhana yaitu data kota seperti tabel di atas. Pertama, kita buat dataframe-nya terlebih dahulu.

import pandas as pd  
  
city\_id = [100, 101, 102, 103, 104, 105, 106]  
city\_name = ['Jakarta', 'Bandung', 'Surabaya', 'Pontianak', 'Medan', 'Makassar', 'Jayapura']  
df = pd.DataFrame(list(zip(city\_id, city\_name)), columns=['ID', 'City'])  
df

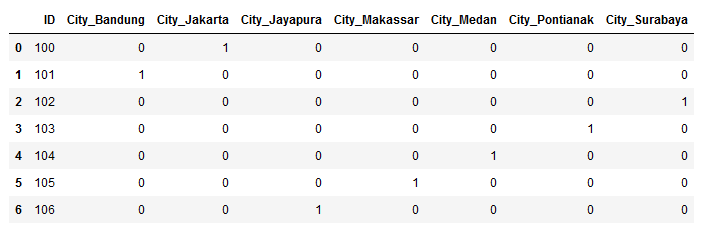
Kode di atas akan mengeluarkan output berupa dataframe df yang memiliki dua kolom yaitu ID dan City seperti di bawah ini.



Kolom City pada dataframe df mengandung nilai bertipe kategori, sehingga kita akan menerapkan metode one-hot encoding menggunakan fungsi **get\_dummies()**yang ada di Pandas. Perhatikan kode berikut.

encoded = pd.get\_dummies(df, prefix='City')  
encoded

Pada atribut **prefix**, kita isikan ‘City’ sebagai awalan dari tiap nama kolom yang di-*encode*. Hasilnya seperti di bawah ini.



**One-Hot Encoding menggunakan Scikit-Learn**

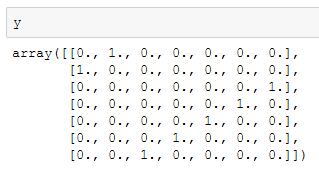
Kita juga dapat menggunakan [OneHotEncoder](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html) dari Scikit-Learn untuk menerapkan one-hot encoding di Python. Perhatikan kode berikut.

from sklearn.preprocessing import OneHotEncoder  
  
y = OneHotEncoder(sparse=False)  
y = y.fit\_transform(df[['City']])  
df\_y = pd.DataFrame(y)

Pertama kita harus import **OneHotEncoder** terlebih dahulu dari *library* **sklearn.preprocessing**. Kemudian kita buat variabel y yang digunakan untuk menampung hasil encoding.

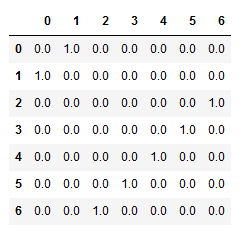
Setelah itu di baris selanjutnya, pada variabel y tersebut dilakukan **fit\_transform()** untuk kolom ‘City’ di dataframe df untuk mengubah nilai kategori pada kolom ‘City’ menjadi angka.

Sampai disini variabel y tersebut masih berupa array.  Jika kita *print* variabel y, hasilnya seperti di bawah ini.



Agar dapat diproses bersama kolom lainnya, variabel y harus dikonversi menjadi dataframe. Pada kode di atas, kita menyimpan dataframe y di dalam variabel df\_y.

Output dari df\_y adalah sebagai berikut.



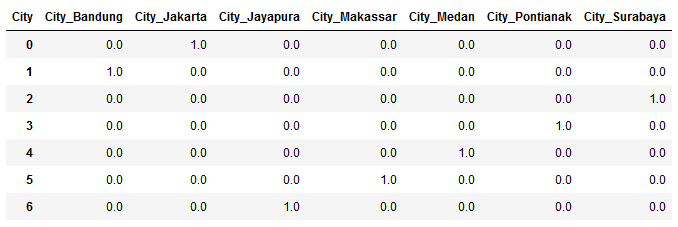
Dari gambar di atas, *header* dataframe hasil one-hot encoding masih berupa bilangan. Untuk dapat memahaminya, kita sebaiknya mengganti *header* tersebut dengan nilai dari df['City'].

Harap dicatat bahwa, hasil one-hot encoding di atas otomatis terurut dari bawah ke atas, atau A – Z, sehingga kita juga harus men-*sorting* df['City'] sebelum dijadikan *header*. Perhatikan kode berikut.

header = df['City'].sort\_values()  
df\_y.columns = 'City\_'+header

Dari kode di atas, df[‘City’] di-*sorting* terlebih dahulu dengan fungsi **sort\_values()** dan dimasukkan ke dalam variabel header. Setelah itu barulah kolom dari dataframe df\_y diganti dengan header yang ditambahkan awalan ‘City\_’ untuk memperjelas bahwa itu adalah hasil *encode* dari kolom ‘City’.

Setelah di-*rename*, output dari dataframe df\_y menjadi seperti di bawah ini.



Setelah itu kita dapat menggabungkannya hasil *encoding* tersebut dengan dataframe df yang sebelumnya agar dapat diproses sebagai satu kesatuan.

df\_join = df.join(df\_y)

Output dari df\_join seperti di bawah ini.

