

# LAPORAN PRAKTIK PREPROSESSING

## DATASET HOUSING

**Nama : Zulfikar Junirto**

**Npm : 22010022**

### A. Source Lengkap

```
import os
import pandas as pd
from sklearn.preprocessing import LabelEncoder, MinMaxScaler

director = []
for i in os.listdir():
    direc_dir = os.path.join(os.getcwd(),i)
    director.append(direc_dir)

print(director)

dataset_housing_train = pd.read_csv(director[1])

dataset_housing_train.head(10)

dataset_housing_train[['LotFrontage','MasVnrArea']].head(10) #menampilkan 10 kolom
LotFrontage dan MasVnrArea

dataset_housing_train[['MSZoning','Neighborhood']].head(10)

#Membuat Objek pada labelImage
dataset_Neighborhood = LabelEncoder()
dataset_MSZoning = LabelEncoder()

# Transformasi kolom 'Neighborhood'
dataset_housing_train['Neighborhood'] =
dataset_Neighborhood.fit_transform(dataset_housing_train['Neighborhood'])

# Transformasi Kolom 'MSZoning'
dataset_housing_train['MSZoning'] =
dataset_MSZoning.fit_transform(dataset_housing_train['MSZoning'])
```

```
dataset_housing_train[['Neighborhood','MSZoning']].head(10)

dataset_housing_train[['SalePrice','GrLivArea']].head(10)

#membuat Objek masing masing Kolom untuk normalisasi
normalisasi_SalePrice = MinMaxScaler()
normalisasi_GrLivArea = MinMaxScaler()

# Melakukan Normalisasi pada Kolom SalePrice
dataset_housing_train['SalePrice'] =
normalisasi_SalePrice.fit_transform(dataset_housing_train[['SalePrice']])
dataset_housing_train['GrLivArea'] =
normalisasi_GrLivArea.fit_transform(dataset_housing_train[['GrLivArea']])

dataset_housing_train[['SalePrice','GrLivArea']].head(10)

dataset_housing_train.to_csv('dataset_housing_train_fix.csv',index=False)
```

## B. Penjelasan Singkat Preprocessing

### 1. Import Library

```
[101] import os
      import pandas as pd
      from sklearn.preprocessing import LabelEncoder, MinMaxScaler
```

Melakukan import Beberapa Library yang digunakan untuk preprocessing Data, yaitu *Pandas*, *Sklearn*, dan *os*.

## 2. Mengatur Director

```
Mengabungkan Director dengan path File

[102] director = []
      for i in os.listdir():
          direc_dir = os.path.join(os.getcwd(),i)
          director.append(direc_dir)

Menampilkan file yang ada di director

[103] print(director)

['/content/.config', '/content/train.csv', '/content/sample_data']
```

Melakukan looping pada Director dan menggabungkan Path dan update kedalam Array Agar mudah untuk Manage file file tertentu.

## 3. Membaca semua Kolom yang ada di file

```
Membaca file dari index ke 2 "train.csv"

[104] dataset_housing_train = pd.read_csv(director[1])

Menampilkan semua kolom dalam rentang 10 baris isi dari dataset Train.csv

dataset_housing_train.head(10)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeatu
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	N
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	N
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	N
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	N
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	N
5	6	50	RL	85.0	14115	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	Sh
6	7	20	RL	75.0	10084	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	N
7	8	60	RL	NaN	10382	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	Sh
8	9	50	RM	51.0	6120	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	N
9	10	190	RL	50.0	7420	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	N

10 rows x 81 columns

Lalu membuat sebuah variabe untuk menampung method `pd_read_csv(FILE DIRECTOR)`, lalu dipanggil lagi variabel tadi juga ditambahkan method `head()` tapi dengan rentang nilai index atau urutannya itu 10.

Note :

Kenapa Outputnya 9 bukan 10? Karna Kelompok lokasi memori berurutan yang semuanya memiliki tipe yang sama. Kumpulan data diindeks, atau diberi nomor, dan dimulai dari 0.

## 4. Missing Value

```
print(dataset_housing_train[['LotFrontage','MasVnrArea']].isnull().sum())

LotFrontage    259
MasVnrArea      8
dtype: int64
```

Menampilkan Jumla data yang kosong kolom `LotFrontage`, dan `MasVnrArea`, mengeluarkan Output dengan `LotFrontage` 250 data dan `MasVnArea` 8 data.

```
dataset_housing_train[['LotFrontage', 'MasVnrArea']].head(10) #menampilkan 10 kolom LotFrontage dan MasVnrArea
```

	LotFrontage	MasVnrArea
0	65.0	196.0
1	80.0	0.0
2	68.0	162.0
3	60.0	0.0
4	84.0	350.0
5	85.0	0.0
6	75.0	186.0
7	NaN	240.0
8	51.0	0.0
9	50.0	0.0

Lalu menampilkan 10 data menggunakan *Method()* dengan rentang 10 untuk melihat mana data yang kosong, dan ternyata ada di bagian *LotFrontage*.

Mengisi Masing masing Kolom yang Sebelumnya Null menggunakan Method *median()* dari pandas dan mengatur 1 angka dibelakang koma dengan method *round()*

```
[115] dataset_housing_train['LotFrontage'] = dataset_housing_train['LotFrontage'].fillna(dataset_housing_train['LotFrontage'].mean()).round(1)
dataset_housing_train['MasVnrArea'] = dataset_housing_train['MasVnrArea'].fillna(dataset_housing_train['MasVnrArea'].mean()).round(1)
```

Lalu Menggunakan *Method mean()* dan *fillna()*, untuk mengisi nilai nilai yang *Nan/atau kosong* lalu menggunakan *method round()* dengan input 1 yang artinya membuat nilai bilangan binner dibelakang koma berjumlah 1 digit.

```
print(dataset_housing_train[['LotFrontage', 'MasVnrArea']].isnull().sum())
dataset_housing_train[['LotFrontage', 'MasVnrArea']].head(10)
```

	LotFrontage	MasVnrArea
0	65.0	196.0
1	80.0	0.0
2	68.0	162.0
3	60.0	0.0
4	84.0	350.0
5	85.0	0.0
6	75.0	186.0
7	70.0	240.0
8	51.0	0.0
9	50.0	0.0

setelah menggunakan method *mean()* lalu menampilkan hasilnya, menggunakan method *isallnum().sum()* akan menampilkan jumlah data yang hilang, lalu dibawahnya juga diperlihatkan menggunakan Method *head(10)* yang artinya menampilkan 10 baris. Dan terlihat pada kolom *LotFrontage()* pada index ke 7 yang tadinya *Nan* menjadi ada nilainya.

## 5. Encoding

### 2. ENCODING Pada Kolom NeighborHoot

menampilkan 10 data dari Kolom "MSZoning", "Neighborhood" yang akan melakukan encoding menggunakan LabelEncoder dari Library Sklearn

```
# dataset = "MSZoning", "Neighborhood"  
dataset_housing_train[['MSZoning', 'Neighborhood']].head(10)
```

	MSZoning	Neighborhood
0	RL	CollgCr
1	RL	Veenker
2	RL	CollgCr
3	RL	Crawfor
4	RL	NoRidge
5	RL	Mitchel
6	RL	Somerst
7	RL	NWArmes
8	RM	OldTown
9	RL	BrkSide

pada proses Encoding ini adalah proses mengubah yang tadinya itu data kategorial menjadi numerik

Melakukan Encoding pada Kolom Mszoning dan Neighborhood karena dalam format numerik agar bisa diproses oleh model

```
# Membuat Objek pada LabelImage  
dataset_Neighborhood = LabelEncoder()  
dataset_MSZoning = LabelEncoder()  
  
# Transformasi kolom 'Neighborhood'  
dataset_housing_train['Neighborhood'] = dataset_Neighborhood.fit_transform(dataset_housing_train['Neighborhood'])  
  
# Transformasi Kolom 'MSZoning'  
dataset_housing_train['MSZoning'] = dataset_MSZoning.fit_transform(dataset_housing_train['MSZoning'])
```

pada source code tersebut yang pertama adalah membuat 2 variabel yang masing masing digunakan untuk per kolom, yang pertama ada *dataset\_Neighborhood* dan *dataset\_MSZoning* yang itu digunakan untuk membuat objek dari library *sklearn* dengan method *labelEncoder* karna nantinya akan otomatis merubah menjadi numerik.

menampilkan Masing Masing Kolom ketika sudah di encoding

```
dataset_housing_train[['Neighborhood', 'MSZoning']].head(10)
```

	Neighborhood	MSZoning
0	5	3
1	24	3
2	5	3
3	6	3
4	15	3
5	11	3
6	21	3
7	14	3
8	17	4
9	3	3

Setelah melakukan proses encoding hasilny akan seperti itu karna dia merubah numerik pada label tertentu misal **RL** adalah angka **1** dan **RM** adalah angka **4**.

## 6. Normalisasi

menampilkan kolom SalePrice dan GrLivArea yang akan di lakukan Normalisasi

```
dataset_housing_train[['SalePrice', 'GrLivArea']].head(10)
```

	SalePrice	GrLivArea
0	208500	1710
1	181500	1262
2	223500	1786
3	140000	1717
4	250000	2198
5	143000	1362
6	307000	1694
7	200000	2090
8	129900	1774
9	118000	1077

menampilkan data sebelum di normalisasi yaitu pada Kolom “*SalePrice*” dan “*GrLivArea*”.

Melakukan Normalisasi Data pada Kolom SalePrice GrLivArea

```
[136] #membuat Objek masing masing Kolom untuk normalisasi
normalisasi_SalePrice = MinMaxScaler()
normalisasi_GrLivArea = MinMaxScaler()

# Melakukan Normalisasi pada Kolom SalePrice
dataset_housing_train['SalePrice'] = normalisasi_SalePrice.fit_transform(dataset_housing_train[['SalePrice']])
dataset_housing_train['GrLivArea'] = normalisasi_GrLivArea.fit_transform(dataset_housing_train[['GrLivArea']])
```

Step ini sama seperti encoding cuma bedanya hanya pada *value* di variabel masing masing kolom yang akan di *Normalisasi*.

```
dataset_housing_train[['SalePrice', 'GrLivArea']].head(10)
```

	SalePrice	GrLivArea
0	0.241078	0.259231
1	0.203583	0.174830
2	0.261908	0.273549
3	0.145952	0.260550
4	0.298709	0.351168
5	0.150118	0.193670
6	0.377864	0.256217
7	0.229274	0.330821
8	0.131926	0.271289
9	0.115401	0.139977

Tampilan setelah kolom *SalePrice* dan *GrLivArea* menjadi bilangan biner yang bertipe data float64. Hal ini akan memudahkan model untuk dilatih.

[Link GitHub Lengkap](#)