

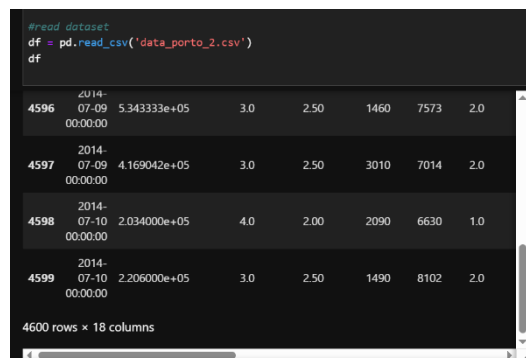
I. BUSSINESS UNDERSTANDING

Properti rumah merupakan asset yang sangat diperlukan oleh setiap orang. Rumah sendiri memiliki pengertian sebagai bangunan yang berfungsi sebagai tempat tinggal layak huni yang dapat melindungi dan merupakan tempat bernaung bagi penghuninya. Selain itu rumah juga memiliki fungsi sebagai asset bagi pemiliknya baik itu secara ekonomi maupun secara emosional. Maka tidak dapat dipungkiri bahwa setiap orang akan sangat memikirkan secara matang apabila akan membeli rumah tersebut.

Namun di masa sekarang banyak orang yang salah memilih untuk membeli rumah yang sesuai dengan keinginan dan kebutuhannya. Tidak sedikit juga orang yang merasa menyesal dan dirugikan baik itu secara fisik, mental, maupun finansial akibat salah memilih properti rumah yang sudah dibeli.

Dengan permasalahan tersebut maka setiap orang yang berencana dan berkeinginan untuk membeli rumah akan memikirkan kemungkinan-kemungkinan baik itu baik maupun buruk yang akan dihadapi apabila sudah membeli rumah tersebut, serta melakukan berbagai perhitungan agar rumah yang telah dibeli dapat menguntungkan bagi diri mereka sendiri.

II. DATA UNDERSTANDING



```
#read dataset
df = pd.read_csv('data_porto_2.csv')
df
```

4596	2014-07-09 00:00:00	5.343333e+05	3.0	2.50	1460	7573	2.0
4597	2014-07-09 00:00:00	4.169042e+05	3.0	2.50	3010	7014	2.0
4598	2014-07-10 00:00:00	2.034000e+05	4.0	2.00	2090	6630	1.0
4599	2014-07-10 00:00:00	2.206000e+05	3.0	2.50	1490	8102	2.0

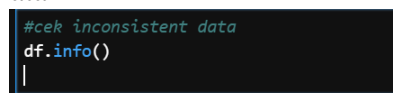
4600 rows x 18 columns

Gambar 1. Hasil membaca data di file.csv

Dataset merupakan file berbentuk .csv yang memiliki 4600 baris serta 18 kolom dengan keterangan kolom yaitu : date, price, bedrooms, bathrooms, sqft_living, sqft_living, floors, waterfront, view, condition, sqft_above, sqft_basement, yr_built, yr_renovated, street, city, statezip, country.

III. DATA PREPARATION

a. Identifikasi Inconsistent Data



```
#cek inconsistent data
df.info()
|
```

Gambar 2. Untuk melihat informasi data di file.csv

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 18 columns):
#   Column             Non-Null Count  Dtype  
---  -
0   date                4600 non-null   object  
1   price               4600 non-null   float64 
2   bedrooms            4600 non-null   float64 
3   bathrooms           4600 non-null   float64 
4   sqft_living         4600 non-null   int64   
5   sqft_lot            4600 non-null   int64   
6   floors              4600 non-null   float64 
7   waterfront          4600 non-null   int64   
8   view                4600 non-null   int64   
9   condition           4600 non-null   int64   
10  sqft_above          4600 non-null   int64   
11  sqft_basement       4600 non-null   int64   
12  yr_built            4600 non-null   int64   
13  yr_renovated        4600 non-null   int64   
14  street              4600 non-null   object  
15  city                4600 non-null   object  
16  statezip            4600 non-null   object  
17  country             4600 non-null   object  
dtypes: float64(4), int64(9), object(5)
memory usage: 647.0+ KB

```

Gambar 3. Hasil data file.csv info

Untuk data di dalam file .csv semua tipe data sudah sesuai dengan isi dari kolom tersebut sehingga dapat disimpulkan tidak ada inconsistent data.

b. Identifikasi Missing Value

Dari Gambar 3. diatas dapat diambil kesimpulan bahwa tidak ada missing value. Hal tersebut dikarenakan untuk semua kolom memiliki nilai non-null sejumlah 4600 yang dimana jumlah baris pada data yaitu 4600.

c. Identifikasi Outlier

```

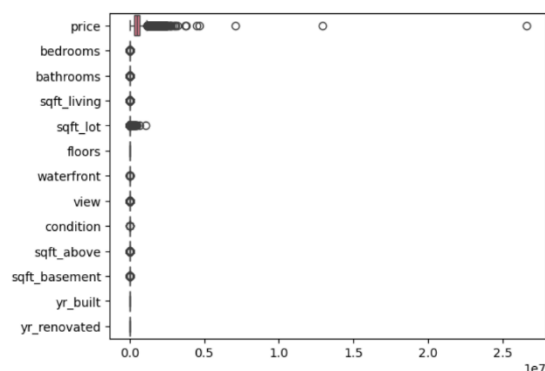
#cek outlier
#mengambil kolom yang bukan tipe data object
non_categorical_columns = df.select_dtypes(include=[np.number]).columns
non_categorical_columns

Index(['price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors',
       'waterfront', 'view', 'condition', 'sqft_above', 'sqft_basement',
       'yr_built', 'yr_renovated'],
      dtype='object')

```

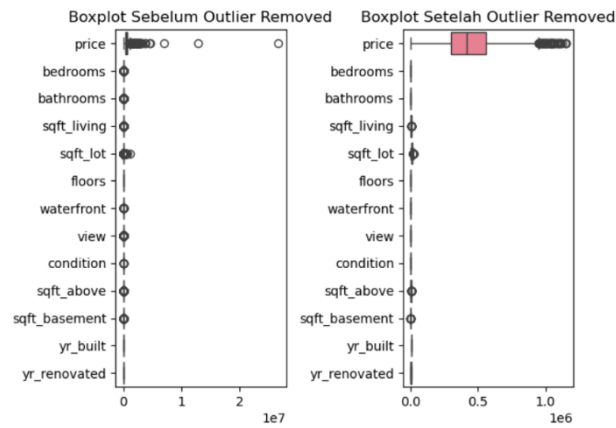
Gambar 4. Filter data yang numerik saja

Untuk identifikasi outlier maka dilakukan filter untuk data yang non numerik terlebih dahulu. Kemudian cek data dengan menggunakan boxplot dan didapatkan hasil seperti gambar 5.



Gambar 5. Hasil boxplot

Dari gambar 5 didapatkan bahwa terdapat outlier dari dataset tersebut, kemudian dilakukan remove outlier agar outlier tersebut hilang. Metode yang digunakan untuk remove outlier yaitu IQR.



Gambar 6. Hasil remove outlier dengan metode IQR

```
<class 'pandas.core.frame.DataFrame'>
Index: 3316 entries, 0 to 4599
Data columns (total 18 columns):
#   Column          Non-Null Count  Dtype
---  -
0   date             3316 non-null   object
1   price            3316 non-null   float64
2   bedrooms         3316 non-null   float64
3   bathrooms        3316 non-null   float64
4   sqft_living      3316 non-null   int64
5   sqft_lot         3316 non-null   int64
6   floors           3316 non-null   float64
7   waterfront       3316 non-null   int64
8   view            3316 non-null   int64
9   condition        3316 non-null   int64
10  sqft_above       3316 non-null   int64
11  sqft_basement    3316 non-null   int64
12  yr_built         3316 non-null   int64
13  yr_renovated     3316 non-null   int64
14  street           3316 non-null   object
15  city             3316 non-null   object
16  statezip         3316 non-null   object
17  country          3316 non-null   object
dtypes: float64(4), int64(9), object(5)
memory usage: 492.2+ KB
```

Gambar 7. Data setelah remove outlier

Setelah dilakukan proses penghapusan untuk outlier dari data maka hasil boxplot yang didapatkan yaitu seperti pada gambar 6. Info data setelah remove outlier dapat dilihat di gambar 7.

d. Identifikasi Duplicate Value

```
#cek data duplikasi dari data yang sudah remove outlier
df_outlier_removed.duplicated(subset=['street'])

0      False
2      False
3      False
4      False
5      False
...
4595   False
4596   False
4597   False
4598   False
4599   False
Length: 3316, dtype: bool
```

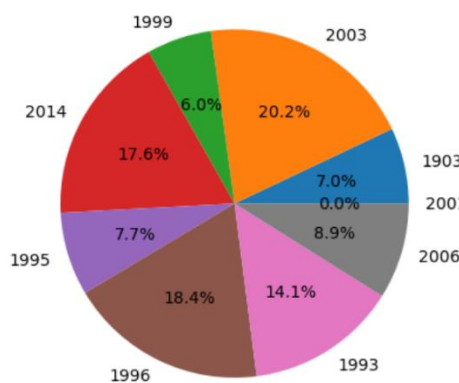
Gambar 8. Cek duplicate value

Setelah dilakukan pengecekan untuk kolom street, tidak ditemukan data duplikasi pada dataset tersebut. Penggunaan kolom street untuk pengecekan dikarenakan data pada kolom street merupakan data yang harus bersifat unik/identik yang tidak boleh ada 2 atau lebih, sedangkan untuk data di kolom lain tidak harus bersifat unik.

IV. KESIMPULAN

Setelah dilakukan identifikasi data dan filter untuk rekomendasi rumah yang akan direkomendasikan. Hasilnya ditemukan inkonsistensi data dimana untuk tahun renovasi lebih kecil dari pada tahun dibangun, yang seharusnya tahun dibangun harus lebih kecil dari tahun renovasi. Sehingga dilakukanlah penghapusan untuk baris tersebut. Filter yang sebelumnya telah dilakukan itu merupakan filter untuk ketentuan yang sesuai dengan syarat yang diajukan serta ada tambahan lain yaitu dengan ketentuan harga lebih kecil dari harga terbesar, kamar mandi dengan nilai terbesar, serta kamar tidur dengan nilai terbesar. Kemudian dilakukan visualisasi dengan membuat pie chart dan barchart.

Rekomendasi Rumah Berdasarkan Harga dan Tahun Dibangun



Gambar 9. Pie chart

Rekomendasi Rumah Berdasarkan Tahun Dibangun dan Tahun Direnovasi

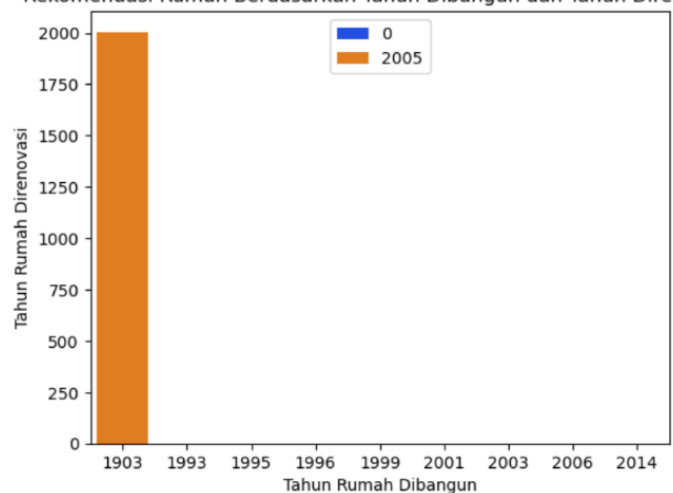


Figure 10. Barchart

Dari hasil pie chart di gambar 9. dapat diambil kesimpulan bahwa rekomendasi rumah yang paling menguntungkan berdasarkan harga dan tahun pembuatan maka berikut list tahunnya : 2001, 1999, dan 1903. Namun apabila dilihat dari barchart pada gambar 10. dapat diambil kesimpulan bahwa hanya ada 1 rumah yang pernah melakukan renovasi yaitu di tahun 1903, sehingga berikut list rekomendasi yang paling menguntungkan yaitu tahun : 1903, 2001, dan 1999.

Sehingga untuk list yang bisa diajukan sebagai rekomendasi rumah yang tepat dan menguntungkan untuk dibeli yaitu rumah yang di jalan:

1	352 17th Ave	6	8921 42nd Ave NE
2	34529 SE Jay Ct	7	15845 SE 44th Ct
3	10026 61st Ave S	8	15563 SE 67th Pl
4	8430 8th Ave SW	9	8309 30th Ave NW
5	114 210th Ave NE		