Prediksi Harga Rumah di California

MEMANFAATKAN DATASET UNTUK PREDIKSI HARGA RUMAH DI CALIFORNIA

Rio Sebastian - 00000026656
Yohanes Wiliam Hadiprojo - 00000063762

Data Analyst

PEMAHAMAN BISNIS DAN DATA

Proyek ini bertujuan untuk memprediksi nilai median rumah di California menggunakan dataset yang disediakan oleh Biro Sensus AS dengan linear regression. Ini penting karena harga rumah adalah indikator penting dari kondisi ekonomi dan dapat membantu dalam membuat keputusan baik oleh bisnis maupun oleh konsumen individu. Dalam konteks keuangan dan perumahan, pemahaman tentang prediksi harga rumah dapat membantu dalam investasi, penentuan nilai pajak, dan kebijakan perencanaan perkotaan.

Dataset ini diambil dari Kaggle

https://www.kaggle.com/datasets/shibumohapatra/house-price

Data
Analyst

DATA PRE-PROCESSING

DATA SET OVERVIEW

```
In [2]: house = pd.read_csv("California House Price.csv")
        house.head(10)
```

Out[2]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity	median_house_value
0	-122.23	37.88	41	880	129.0	322	126	8.3252	NEAR BAY	452600
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	NEAR BAY	358500
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	NEAR BAY	352100
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	NEAR BAY	341300
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	NEAR BAY	342200
5	-122.25	37.85	52	919	213.0	413	193	4.0368	NEAR BAY	269700
6	-122.25	37.84	52	2535	489.0	1094	514	3.6591	NEAR BAY	299200
7	-122.25	37.84	52	3104	687.0	1157	647	3.1200	NEAR BAY	241400
8	-122.26	37.84	42	2555	665.0	1206	595	2.0804	NEAR BAY	226700
9	-122.25	37.84	52	3549	707.0	1551	714	3.6912	NEAR BAY	261100

Dataset ini berisi longitude, latitude, housing_median_age, total_rooms, total_bedrooms, population, households, median_income, ocean_proximity, dan median_house_value.

DATA PREPARATION

house.info() digunakan untuk menilai dengan cepat struktur kumpulan data kami, termasuk tipe data dan nilai yang hilang, memastikan dasar yang kuat untuk analisis.

```
In [3]: house.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  int64  
 3   total_rooms         20640 non-null  int64  
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  int64  
 6   households          20640 non-null  int64  
 7   median_income       20640 non-null  float64
 8   ocean_proximity     20640 non-null  object 
 9   median_house_value  20640 non-null  int64  
dtypes: float64(4), int64(5), object(1)
memory usage: 1.6+ MB
```

```
In [4]: house.describe()
```

Out[4]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

```
In [5]: house.isna().sum()

Out[5]: longitude             0
        latitude              0
        housing_median_age    0
        total_rooms           0
        total_bedrooms      207
        population            0
        households            0
        median_income         0
        ocean_proximity       0
        median_house_value    0
        dtype: int64
```

disini dilihat bahwa total_bedrooms memiliki 207 na, maka dari itu di filling missing valuenya

```
In [5]: # Handling missing values
        house['total_bedrooms'].fillna(house['total_bedrooms'].median(), inplace=True)
```

mengisi(filling) missing values di 'total_bedrooms' column dengan median value dari *column*-nya.

```
In [6]: # Kolom yang akan ditransformasi logaritmik
        logtransform = ['total_rooms', 'total_bedrooms', 'population', 'households', 'median_income']
```

lalu ini kolom yang nantinya diproses transformasi logaritmik, yaitu ada total_rooms, total_bedrooms, population, households, dan median_income untuk membuat distribusi data lebih mendekati distribusi normal

Normalisasi

In [8]:
```
# Normalization using Min-Max Scaling
scaler = MinMaxScaler()
numerical_cols = house.select_dtypes(include=['float64', 'int64']).columns
house[numerical_cols] = scaler.fit_transform(house[numerical_cols])
```

Encoding

In [9]:
```
# Encoding for categorical data using one-hot encoding
house_encoded = pd.get_dummies(house, columns=['ocean_proximity'])
```

Mengecek nilai yang kosong (NaN/Not a number) di dalam dataframe house_encoded

In [10]:
```
# Cek apakah ada nilai NaN dalam data
print("Cek nilai NaN sebelum pemodelan:")
print(house_encoded.isna().sum())
```

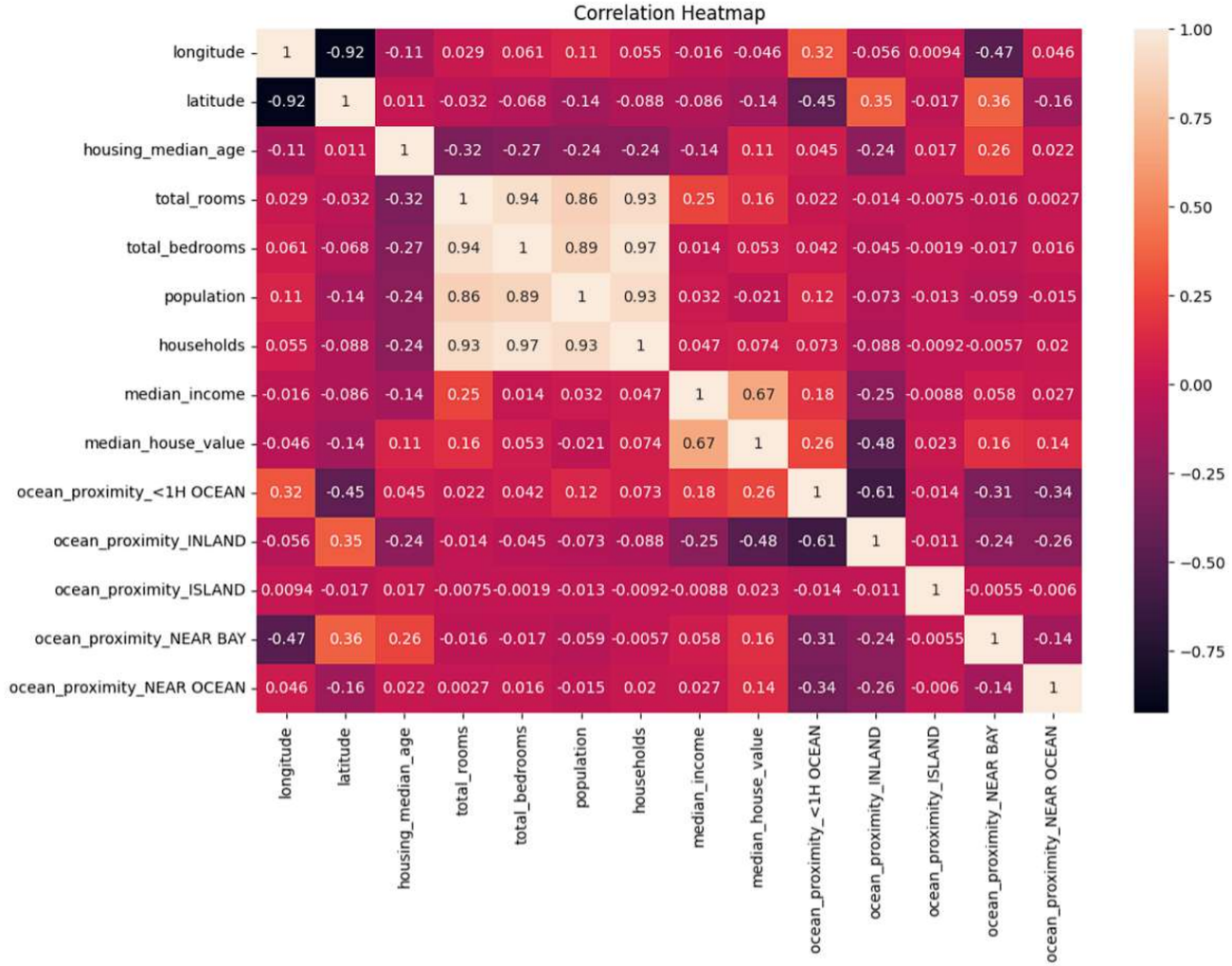
```
Cek nilai NaN sebelum pemodelan:
longitude                   0
latitude                    0
housing_median_age          0
total_rooms                 0
total_bedrooms              0
population                  0
households                  0
median_income               0
median_house_value          0
ocean_proximity_<1H OCEAN   0
ocean_proximity_INLAND      0
ocean_proximity_ISLAND      0
ocean_proximity_NEAR BAY    0
ocean_proximity_NEAR OCEAN  0
dtype: int64
```

DATA VISUALIZATION

VISUALISASI HEATMAP

```
In [13]: # Correlation Heatmap
         plt.figure(figsize=(12, 8))
         sns.heatmap(house_encoded.corr(), annot=True)
         plt.title('Correlation Heatmap')
```

```
Out[13]: Text(0.5, 1.0, 'Correlation Heatmap')
```

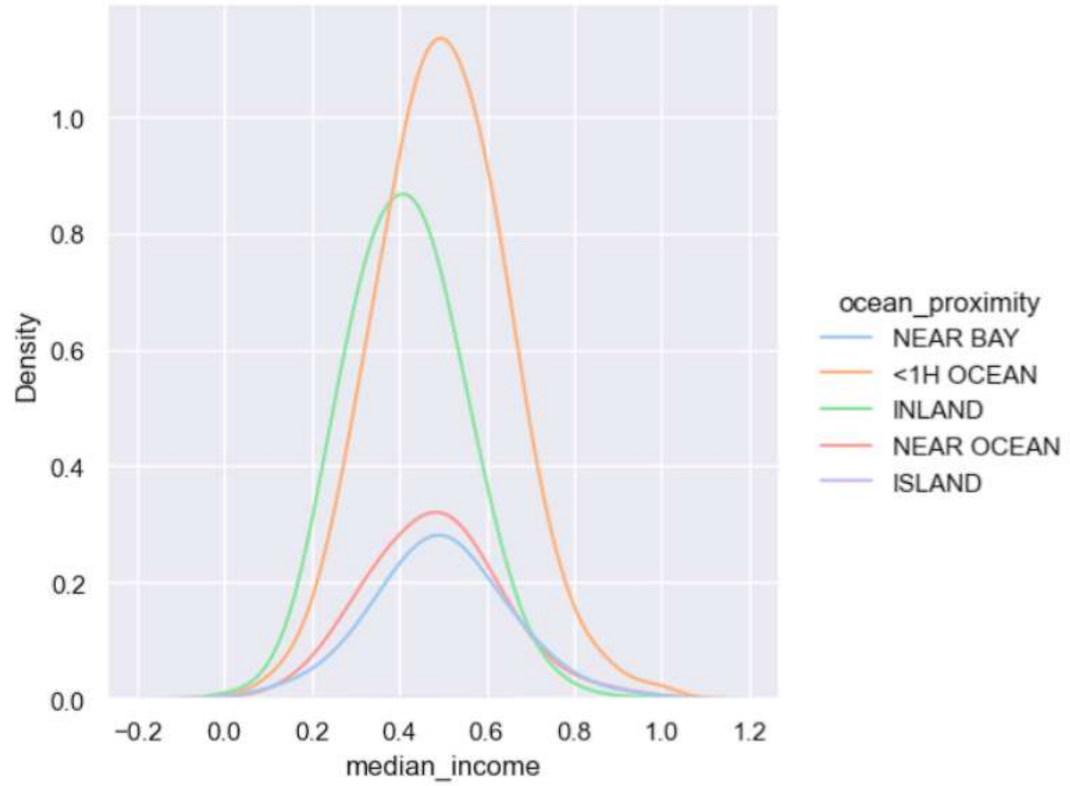


Heatmap korelasi yang disediakan mengungkapkan hubungan dalam kumpulan data harga rumah. Korelasi positif meliputi total kamar dan kamar tidur, total kamar dan rumah tangga, median pendapatan dan nilai rumah, serta kedekatan laut (NEAR BAY) dengan (NEAR OCEAN). Korelasi negatif melibatkan usia median perumahan dengan kamar tidur, rumah tangga, pendapatan median, dan nilai rumah, yang menunjukkan hubungan intuitif.

VISUALISASI PLOT KDE

```
In [12]: sns.set(rc={'figure.figsize': (20, 8)})
         sns.displot(data=house, x='median_income', hue='ocean_proximity', kind="kde", palette="pastel", bw_adjust=2)
```

```
Out[12]: <seaborn.axisgrid.FacetGrid at 0x1c8c977eb10>
```



Berdasarkan plot KDE yang dibuat, semakin besar gajinya, semakin dekat rumahnya menuju lautan.

Ini dikarenakan rumah yang berlokasi di dekat tepi pantai lebih mahal dan lebih disukai.

VISUALISASI SCATTER PLOT

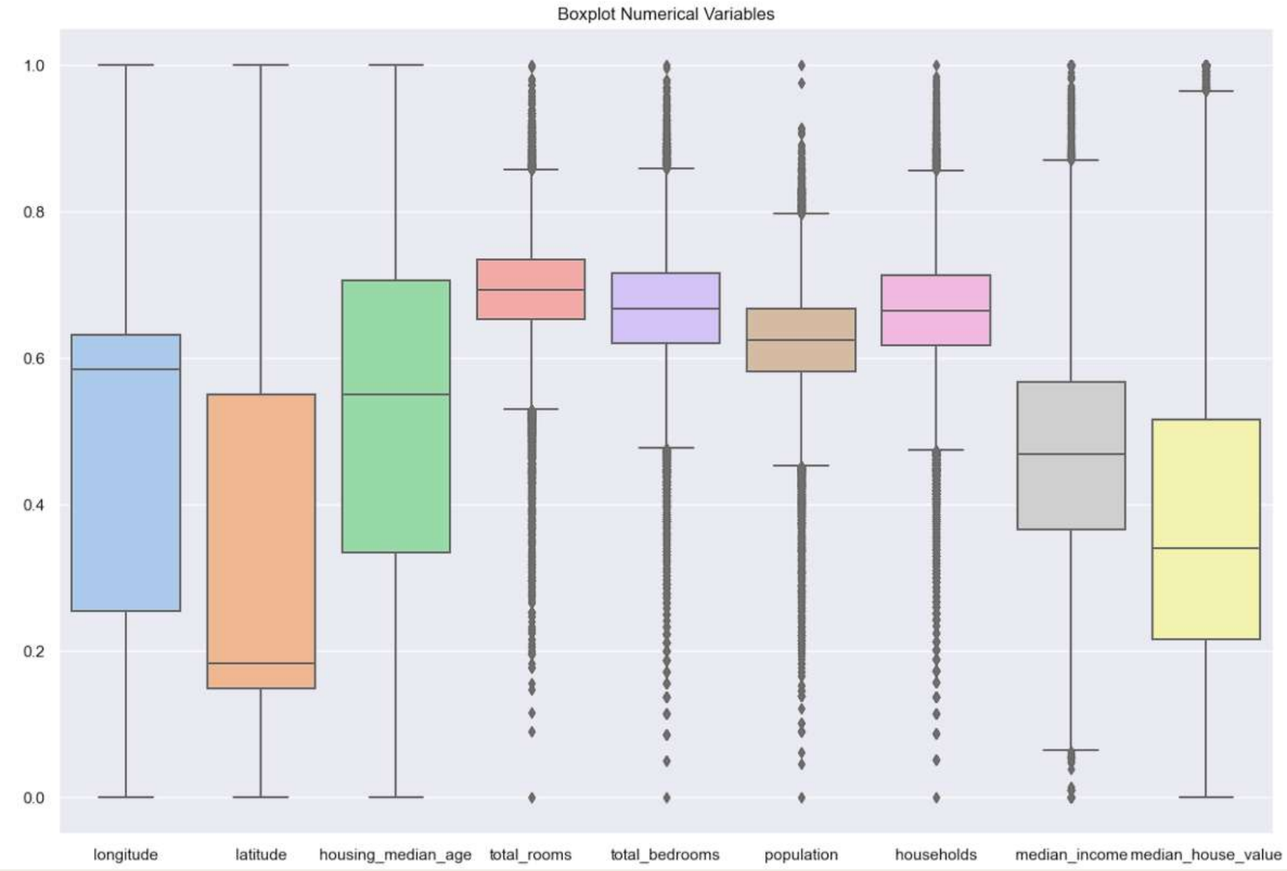
In [13]:
```
# Scatter Plot of Median Income vs Median House Value
sns.scatterplot(data=house,
                x="median_income",
                y="median_house_value",
                hue="ocean_proximity",
                palette="pastel")
```

Out[13]: <Axes: xlabel='median_income', ylabel='median_house_value'>

hubungan antara pendapatan median dan nilai rumah median dalam kumpulan data perumahan. Hal ini menunjukkan bahwa terdapat korelasi positif antara kedua variabel dan ada beberapa variasi dalam hubungan tersebut tergantung pada kedekatan laut.

VISUALISASI BOX AND WHISKER PLOT

In [14]:
```
# Boxplot for all numerical columns after preprocessing
plt.figure(figsize=(15, 10))
sns.boxplot(data=house_encoded[numerical_cols], palette="pastel")
plt.title('Boxplot Numerical Variables')
plt.show()
```

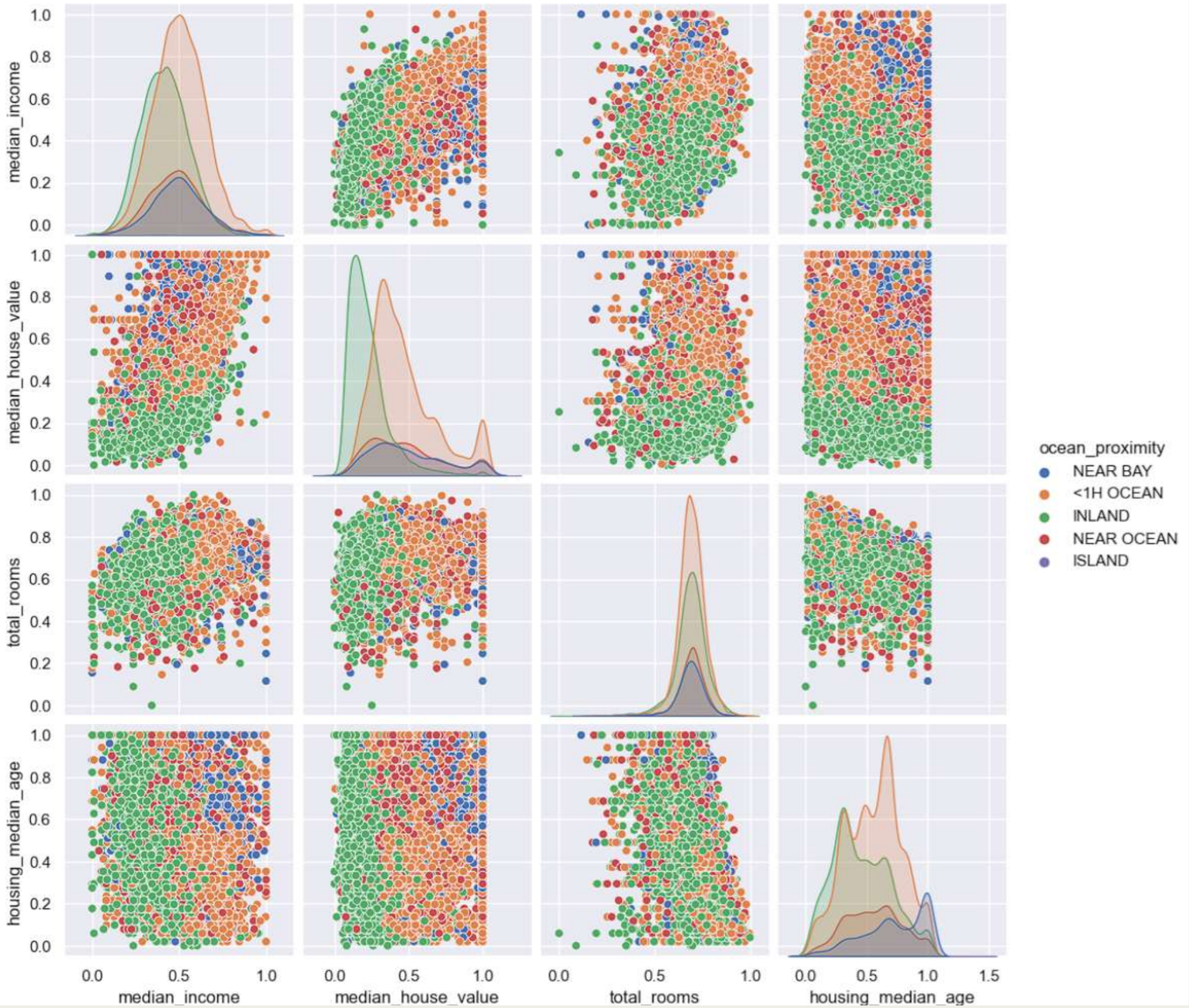


Membuat boxplot untuk semua variabel numerik setelah pra-pemrosesan untuk menilai distribusi dan adanya outlier.

distribusi variabel numerik dalam dataset harga rumah.

VISUALISASI PAIRPLOT

```
In [15]: # Pairplot for selected columns
sns.pairplot(house, vars=['median_income', 'median_house_value', 'total_rooms', 'housing_median_age'], hue='ocean_proximity')
plt.show()
```



Pairplot ini menampilkan hubungan pairwise antara variabel median_income, median_house_value, total_rooms, dan housing_median_age, dengan pewarnaan berbeda (hue) berdasarkan ocean_proximity. Ini membantu visualisasi dan memahami bagaimana variabel-variabel ini saling berinteraksi dan berkorelasi satu sama lain dalam konteks lokasi perumahan.

DATA MODELLING

PEMODELAN

menyiapkan data untuk model regresi linear, mencari variable dependen(y = harga rumah) dari independen (X = ocean_proximity)

In [16]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Prepare the features and target variable
X = house_encoded.drop('median_house_value', axis=1)
y = house_encoded['median_house_value']
```

menyiapkan data untuk training dan evaluasi machine learning model

In [17]:
```
# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

memulai Linear Regression model lalu fitting ke training data

In [18]:
```
# Initialize the Linear Regression model
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)
```

Out[18]:
```
▾ LinearRegression
LinearRegression()
```

EVALUASI MODELING

predictions untuk testing set menggunakan Mean Squared Error (MSE)

```
In [19]: # Membuat prediksi dengan model
         y_pred = linear_model.predict(X_test)
```

```
In [20]: # Menghitung metrik evaluasi
         mse = mean_squared_error(y_test, y_pred)
         r2 = r2_score(y_test, y_pred)
         print(f"Mean Squared Error: {mse}")
         print(f"R-squared: {r2}")

         Mean Squared Error: 0.022342802013355114
         R-squared: 0.5989321627075603
```

MSE menunjukkan kinerja prediksi yang baik, dan nilai spesifik 0,0223 menunjukkan bahwa, secara rata-rata, prediksi model mendekati nilai aktual.

Nilai R-squared sebesar 0,5989 menunjukkan bahwa model tersebut menjelaskan hampir 60% variabilitas harga rumah dalam variabel target. Hal ini menunjukkan kecocokan yang cukup baik, menangkap sebagian besar variabilitas yang diamati.

menghitung error diantara nilai target sebenarnya (y_test) dengan nilai yang diprediksi (y_pred) untuk visualisasi

```
In [22]: # Hitung error
         error = y_test - y_pred
```

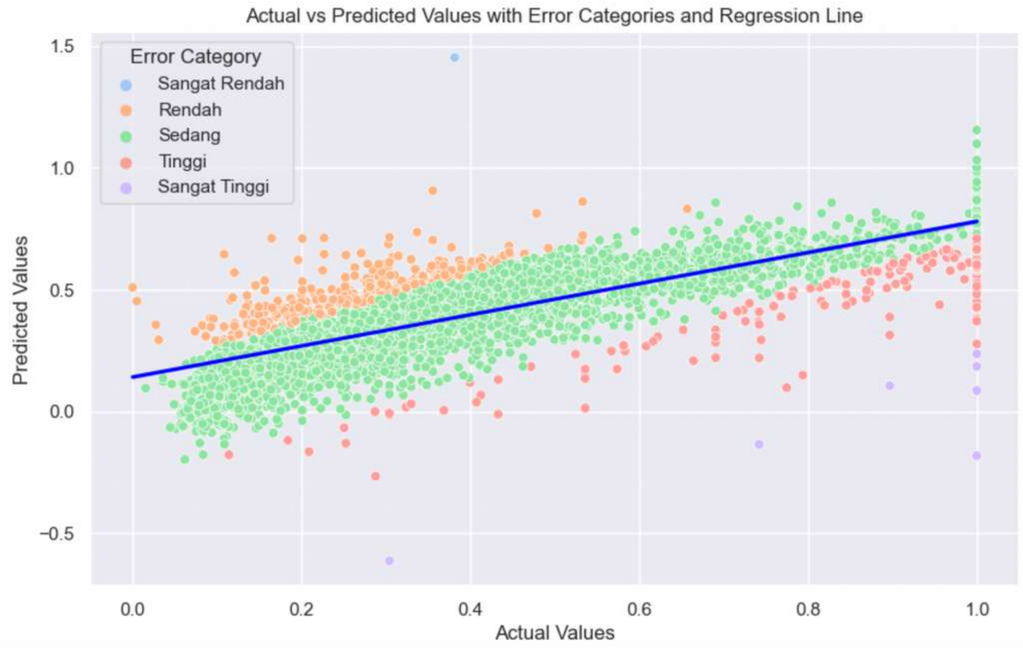
```
In [23]: # Kategorisasi error
         error_categories = pd.cut(error, bins=5, labels=['Sangat Rendah', 'Rendah', 'Sedang', 'Tinggi', 'Sangat Tinggi'])
```

```
In [24]: # Buat DataFrame untuk visualisasi
         plot_data = pd.DataFrame({'Actual Values': y_test, 'Predicted Values': y_pred, 'Error Category': error_categories})
```

```
In [32]: plt.figure(figsize=(10, 6))
sns.scatterplot(data=plot_data, x='Actual Values', y='Predicted Values', hue='Error Category', palette='pastel')

# Menambahkan garis regresi linear pada scatter plot yang sama
sns.regplot(x='Actual Values', y='Predicted Values', data=plot_data, scatter=False, color='blue')

plt.title("Actual vs Predicted Values with Error Categories and Regression Line")
plt.show()
```



Garis trendnya menunjukan bahwa ketepatan modelnya berada di error category sedang.

Scatterplot ini menampilkan perbandingan antara nilai sebenarnya (Actual Values) dan nilai prediksi (Predicted Values) dari model regresi linier. Scatterplot tersebut juga menggunakan Error Category sebagai hue untuk memberikan informasi tambahan mengenai kategori kesalahan (error) dalam prediksi.

Ketepatan Model: Visualisasi menunjukkan bahwa model memiliki tingkat ketepatan yang wajar dalam memprediksi harga rumah, dengan banyak titik data yang berdekatan dengan garis tren.

Terima Kasih