

DOKUMEN PROYEK

12S3205 - PENAMBANGAN DATA

Development of a Predictive Regression Model for House Prices Using Ensemble Stacking Techniques

Disusun Oleh:

12S22015	Angelina Nadeak
12S22029	Jeremy Samosir
12S22038	Ade Siahaan
12S22052	Rosari Simanjuntak



PROGRAM STUDI SARJANA SISTEM INFORMASI

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO (FITE)
INSTITUT TEKNOLOGI DEL
TAHUN 2024/2025**

DAFTAR ISI

BAB 1 BUSINESS UNDERSTANDING	3
1.1 Determine Business Objective	3
1.2 Determine Project Goal	3
1.3 Produce Project Plan	4
BAB 2 DATA UNDERSTANDING	5
2.1 Collecting Data	5
Gambar 1 jumlah Training set dan Testing set	5
2.2 Describe Data	5
2.3 Validation Data	7
BAB 3 DATA PREPARATION	8
3.1 Data Selection	8
3.2 Data Cleaning	9
3.3 Data Construct	9
3.4 Labeling Data	9
3.5 Data Integration	9
BAB 4 MODELLING	10
4.1 Build Model	10
BAB 5 EVALUATION	11
BAB 6 DEPLOYMENT	12
DAFTAR PUSTAKA	13

BAB 1 BUSINESS UNDERSTANDING

1.1 Determine Business Objective

Industri properti merupakan salah satu sektor yang sangat dinamis dan memiliki dampak signifikan terhadap perekonomian suatu negara. Harga rumah menjadi indikator utama dalam transaksi jual-beli properti, baik untuk konsumen perorangan, agen properti, maupun perusahaan pengembang real estate. Namun, penentuan harga rumah seringkali bersifat subjektif dan sangat dipengaruhi oleh faktor-faktor eksternal yang sulit diprediksi seperti kondisi pasar, lokasi, dan tren ekonomi.

Tujuan bisnis dari proyek ini adalah menyediakan sistem prediksi harga rumah berbasis machine learning yang mampu mengurangi ketidakpastian dalam proses estimasi harga. Dengan sistem prediksi ini, diharapkan stakeholder properti seperti investor, penjual, pembeli, dan agen real estate dapat mengambil keputusan yang lebih cepat dan tepat.

Manfaat yang ingin dicapai dalam jangka panjang:

- Meminimalisir kesalahan estimasi harga.
- Memberikan insight berbasis data dalam proses negosiasi properti.
- Meningkatkan efisiensi waktu dan biaya dalam menentukan harga jual/beli.
- Meningkatkan daya saing perusahaan properti melalui adopsi teknologi AI.

1.2 Determine Project Goal

Tujuan teknis dari proyek ini adalah mengembangkan model prediksi harga rumah dengan pendekatan ensemble stacking, yang menggabungkan kekuatan dari beberapa algoritma machine learning seperti XGBoost, LightGBM, dan CatBoost. Model stacking ini bertujuan memaksimalkan akurasi prediksi dan meminimalkan error, terutama dalam kondisi data yang kompleks dan beragam.

Target pengembangan model:

- Memprediksi harga rumah berdasarkan fitur properti dengan akurasi tinggi.
- Mengurangi bias prediksi yang disebabkan oleh model tunggal.
- Menghasilkan model yang stabil dan generalisasi dengan baik terhadap data baru.

1.3 Produce Project Plan

Rencana pelaksanaan proyek:

- Tahap 1: Pengumpulan dan eksplorasi data properti.
- Tahap 2: Preprocessing, feature selection dan feature engineering.
- Tahap 3: Pengembangan model ensemble stacking.
- Tahap 4: Evaluasi model menggunakan beberapa metrik evaluasi, di antaranya:
 - RMSE (Root Mean Squared Error) untuk menghitung akar rata-rata kesalahan kuadrat prediksi.
 - MAE (Mean Absolute Error) untuk mengukur rata-rata selisih absolut antara nilai prediksi dan nilai aktual.
 - R^2 (Coefficient of Determination) untuk menilai seberapa besar variasi target yang dapat dijelaskan oleh model.
 - MAPE (Mean Absolute Percentage Error) untuk mengetahui rata-rata kesalahan dalam bentuk persentase.
- Tahap 5: Deployment model dalam bentuk aplikasi prediktif.

Waktu estimasi pengerjaan: 2 bulan
Tim pelaksana: Data Scientist, Data Engineer, Business Analyst.

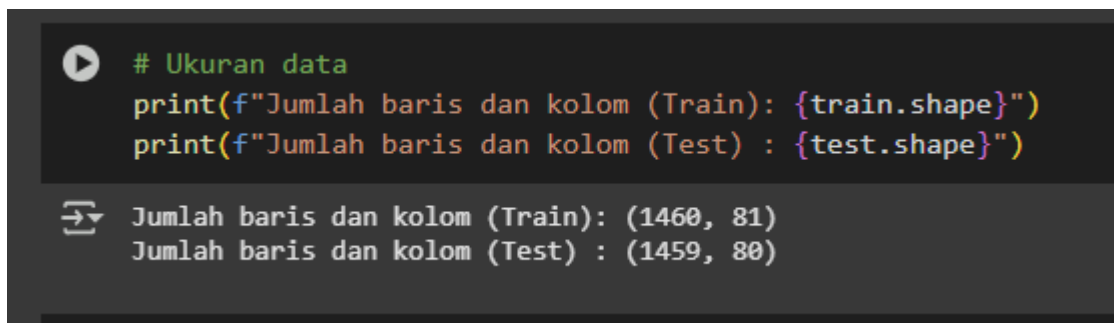
BAB 2 DATA UNDERSTANDING

2.1 Collecting Data

Dataset yang digunakan pada proyek ini diambil dari kompetisi "House Prices - Advanced Regression Techniques" di platform Kaggle. Dataset ini sangat kaya dan telah menjadi benchmark umum dalam pengembangan model regresi prediksi harga properti.

Jumlah data:

- Training set: 1.460 baris data.
- Testing set: 1.459 baris data.
- Total fitur: 81 fitur.



```
# Ukuran data
print(f"Jumlah baris dan kolom (Train): {train.shape}")
print(f"Jumlah baris dan kolom (Test) : {test.shape}")

Jumlah baris dan kolom (Train): (1460, 81)
Jumlah baris dan kolom (Test) : (1459, 80)
```

Gambar 1 jumlah training set dan testing set

Data yang dikumpulkan mencakup berbagai aspek properti, antara lain:

- Karakteristik fisik rumah (luas bangunan, jumlah kamar tidur, jumlah kamar mandi, tahun pembangunan).
- Kualitas properti (rating konstruksi, kondisi bangunan).
- Lokasi properti (nama lingkungan, jarak ke fasilitas umum).
- Fitur eksternal (kondisi halaman, jenis garasi, tipe atap).

2.2 Describe Data

Setelah pengumpulan data, proses berikutnya adalah memahami distribusi dan karakteristik data. Dataset ini mencakup:

- Describe Data Train

code:

```
train.describe()
```

output:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	...	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SeasonPorch	ScreenPorch	PoolArea	MiscVal	NoSold	YrSold	SalePrice
count	1460.000000	1460.000000	1281.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452.000000	1460.000000	...	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000
mean	730.500000	56.897260	70.049958	10516.820882	6.099315	5.575342	1971.267808	1984.865753	103.685262	443.639726	...	84.244521	46.660274	21.954110	3.409589	15.060959	2.758864	43.489041	6.321918	2007.815753	180321.195890
std	421.610009	42.309571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.066207	456.098091	...	125.338794	66.256028	61.119149	29.317331	55.757415	40.177307	496.123024	2.703628	1.328095	79442.502883
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	2006.000000	34900.000000
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	5.000000	2007.000000	129975.000000
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.000000	383.500000	...	0.000000	25.000000	0.000000	0.000000	0.000000	0.000000	0.000000	6.000000	2008.000000	163000.000000
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	166.000000	712.250000	...	168.000000	68.000000	0.000000	0.000000	0.000000	0.000000	0.000000	8.000000	2009.000000	214000.000000
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	...	857.000000	547.000000	552.000000	508.000000	480.000000	738.000000	15500.000000	12.000000	2010.000000	755000.000000

8 rows × 38 columns

- Fitur numerik dan kategorikal :
 - a. Fitur numerik: LotArea, GrLivArea, YearBuilt, TotalBsmtSF, GarageArea.
 - b. Fitur kategorikal: Neighborhood, HouseStyle, ExterQual, GarageType.

Code:

```
numeric_features = train.select_dtypes(include=['int64', 'float64']).columns.tolist()
print("Fitur Numerik:")
print(numeric_features)

categorical_features = train.select_dtypes(include=['object']).columns.tolist()
print("\nFitur Kategorikal:")
print(categorical_features)

print(f"\nJumlah fitur numerik: {len(numeric_features)}")
print(f"Jumlah fitur kategorikal: {len(categorical_features)}")
```

Gambar 2 jumlah fitur numerik dan fitur kategorikal

Output:

Fitur Numerik:

```
['Id', 'MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual',
'OverallCond', 'YearBuilt', 'YearRemodAdd', 'MasVnrArea',
'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF',
'1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea',
'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces',
'GarageYrBlt', 'GarageCars', 'GarageArea', 'WoodDeckSF',
'OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch',
'PoolArea', 'MiscVal', 'MoSold', 'YrSold', 'SalePrice']
```

Fitur Kategorikal:

```
['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour',
'Utilities', 'LotConfig', 'LandSlope', 'Neighborhood',
'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd',
'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation',
'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1',
'BsmtFinType2', 'Heating', 'HeatingQC', 'CentralAir',
'Electrical', 'KitchenQual', 'Functional', 'FireplaceQu',
'GarageType', 'GarageFinish', 'GarageQual', 'GarageCond',
'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature', 'SaleType',
'SaleCondition']
```

Jumlah fitur numerik: 38

Jumlah fitur kategorikal: 43

output dalam bentuk tabel

Fitur Numerik	Fitur Kategorikal
Id	MSZoning
MSSubClass	Street
LotFrontage	Alley
LotArea	LotShape
OverallQual	LandContour
OverallCond	Utilities
YearBuilt	LotConfig
YearRemodAdd	LandSlope
MasVnrArea	Neighborhood

BsmtFinSF1	Condition1
BsmtFinSF2	Condition2
BsmtUnfSF	BldgType
TotalBsmtSF	HouseStyle
1stFlrSF	RoofStyle
2ndFlrSF	RoofMatl
LowQualFinSF	Exterior1st
GrLivArea	Exterior2nd
BsmtFullBath	MasVnrType
BsmtHalfBath	ExterQual
FullBath	ExterCond
HalfBath	Foundation
BedroomAbvGr	BsmtQual
KitchenAbvGr	BsmtCond
TotRmsAbvGrd	BsmtExposure
Fireplaces	BsmtFinType1
GarageYrBlt	BsmtFinType2
GarageCars	Heating
GarageArea	HeatingQC
WoodDeckSF	CentralAir
OpenPorchSF	Electrical
EnclosedPorch	KitchenQual
3SsnPorch	Functional
ScreenPorch	FireplaceQu

PoolArea	GarageType
MiscVal	GarageFinish
MoSold	GarageQual
YrSold	GarageCond
SalePrice	PavedDrive
	PoolQC
	Fence
	MiscFeature
	SaleType
	SaleCondition

Sebagian besar variabel numerik memiliki distribusi yang skewed, terutama variabel SalePrice sebagai label target, yang menunjukkan kecenderungan outlier pada rumah-rumah mewah. Oleh karena itu, analisis statistik deskriptif seperti mean, median, standar deviasi, serta visualisasi boxplot dan histogram dilakukan untuk memahami sebaran data.

- Fitur yang memiliki missing value:

Dalam dataset ini, terdapat 19 fitur yang memiliki nilai kosong (missing values). Fitur-fitur tersebut meliputi PoolQC, MiscFeature, Alley, Fence, MasVnrType, MasVnrArea, FireplaceQu, LotFrontage, GarageType, GarageYrBlt, GarageFinish, GarageQual, GarageCond, BsmtExposure, BsmtFinType1, BsmtFinType2, BsmtQual, BsmtCond, Electrical dan total nilai kosong dari semua fitur ini berjumlah 9.930 data poin.

Code:

```
# Melihat nilai yang hilang (missing values)
print("\nFitur yang memiliki nilai kosong pada Train:")
missing_train = train.isnull().sum()
missing_train = missing_train[missing_train > 0].sort_values(ascending=False)
display(missing_train)
```

Output:

Fitur	Missing Value
PoolQC	1453
MiscFeature	1406
Alley	1369
Fence	1179
MasVnrType	872
FireplaceQu	690
LotFrontage	259
GarageType	81
GarageYrBlt	81
GarageFinish	81
GarageQual	81
GarageCond	81
BsmtExposure	38
BsmtFinType2	38
BsmtQual	37
BsmtCond	37
BsmtFinType1	37
MasVnrArea	8
Electrical	1
Total	7829

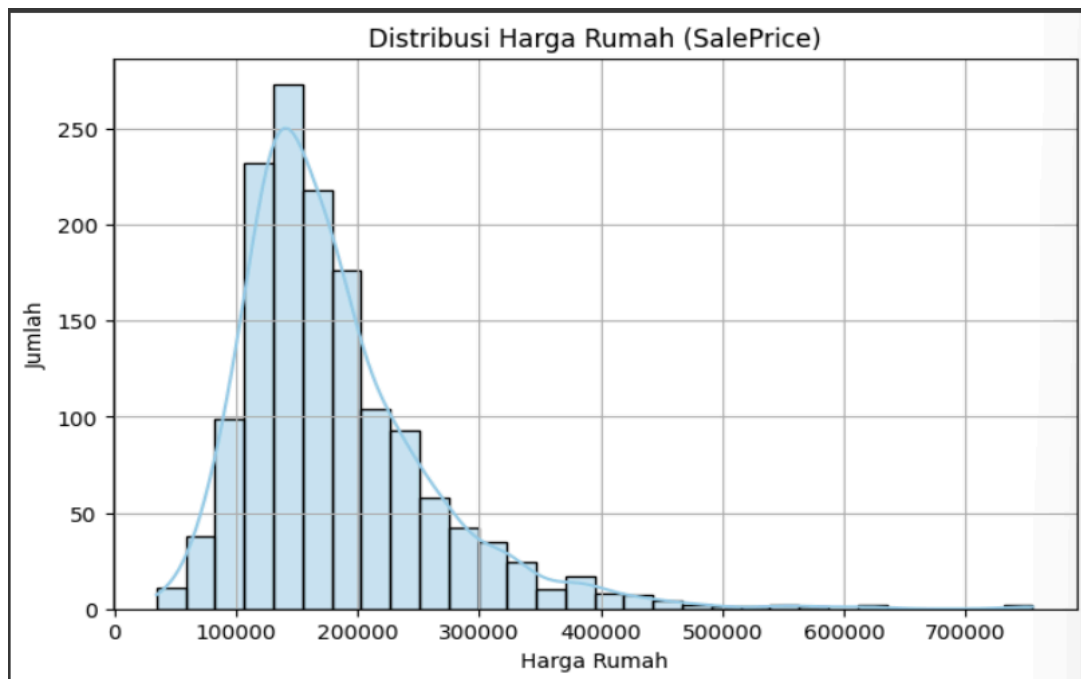
- Visualisasi Distribusi Sale Price (target):

Visualisasi Distribusi *Sale Price* (target) untuk memahami distribusi harga penjualan

Code:

```
# Visualisasi distribusi harga rumah (target)
plt.figure(figsize=(8, 5))
sns.histplot(train['SalePrice'], kde=True, bins=30, color='skyblue')
plt.title('Distribusi Harga Rumah (SalePrice)')
plt.xlabel('Harga Rumah')
plt.ylabel('Jumlah')
plt.grid(True)
plt.show()
```

Output:



Interpretasi:

Visualisasi ini menunjukkan distribusi harga rumah (SalePrice) dalam bentuk histogram dengan kurva KDE. Hasilnya memperlihatkan bahwa distribusi data miring ke kanan (positively skewed), artinya sebagian besar rumah memiliki harga yang relatif rendah, sementara hanya sedikit rumah yang memiliki harga sangat tinggi. Puncak distribusi berada di kisaran 100.000 hingga 200.000, yang merupakan rentang harga paling umum dalam dataset.

- Heatmap korelasi antar fitur numerik

code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder

# Salin dataset
data_corr = train.copy()

# Pisahkan fitur numerik dan kategorikal
numerical_cols = data_corr.select_dtypes(include=['int64', 'float64']).columns.tolist()
categorical_cols = data_corr.select_dtypes(include=['object', 'category']).columns.tolist()

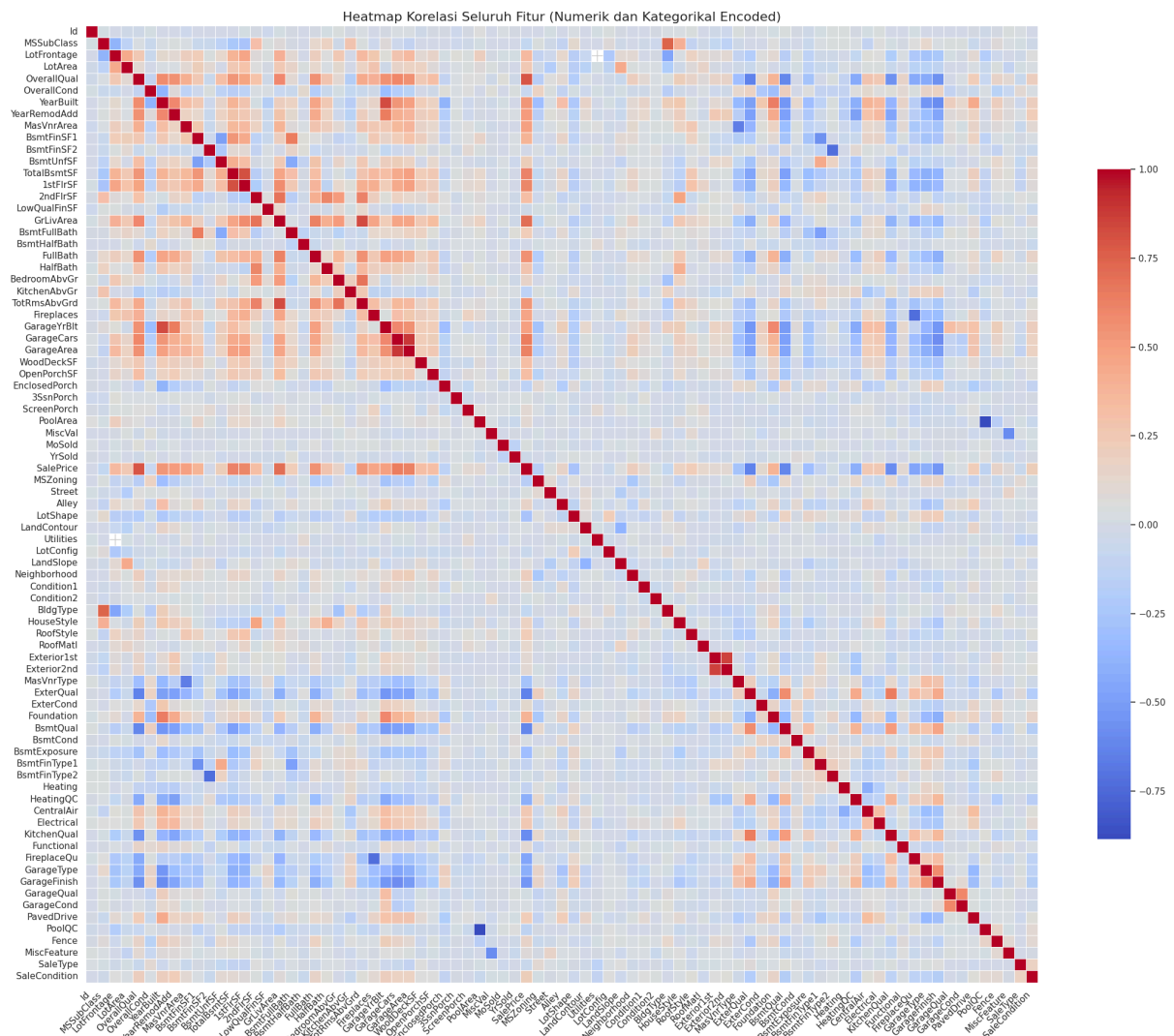
# Encode fitur kategorikal
le = LabelEncoder()
for col in categorical_cols:
    data_corr[col] = le.fit_transform(data_corr[col].astype(str))

# Gabungkan semua fitur numerik (asli + hasil encoding)
all_numeric = data_corr[numerical_cols + categorical_cols]

# Hitung korelasi antar fitur
correlation_matrix = all_numeric.corr(method='pearson')

# Visualisasi heatmap
plt.figure(figsize=(22, 18))
sns.heatmap(
    correlation_matrix,
    cmap='coolwarm',
    annot=False,      # Bisa diganti True jika ingin angka korelasi terlihat
    fmt=".2f",
    linewidths=0.5,
    cbar_kws={"shrink": 0.7}
)
plt.title("Heatmap Korelasi Seluruh Fitur (Numerik dan Kategorikal Encoded)", fontsize=16)
plt.xticks(rotation=45, ha='right')
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```

Output:



Interpretasi:

Beberapa variabel numerik memiliki hubungan yang cukup kuat satu sama lain. Variabel seperti OverallQual, GrLivArea, GarageCars, dan GarageArea menunjukkan korelasi positif yang tinggi terhadap SalePrice, yang berarti semakin baik kualitas bangunan, semakin luas area hunian, dan semakin besar kapasitas garasi, maka harga rumah cenderung semakin mahal. Selain itu, ada juga hubungan kuat antar fitur yang saling berkaitan secara logis, seperti GarageCars dengan GarageArea, serta 1stFlrSF dengan TotalBsmtSF, yang menunjukkan bahwa rumah dengan lantai dasar luas cenderung memiliki basement yang luas juga. Sementara itu, variabel seperti Id, MoSold, YrSold, dan MiscVal tampak memiliki korelasi yang sangat rendah atau tidak signifikan dengan variabel lain, sehingga kemungkinan kurang relevan dalam memengaruhi harga rumah. Secara keseluruhan, heatmap

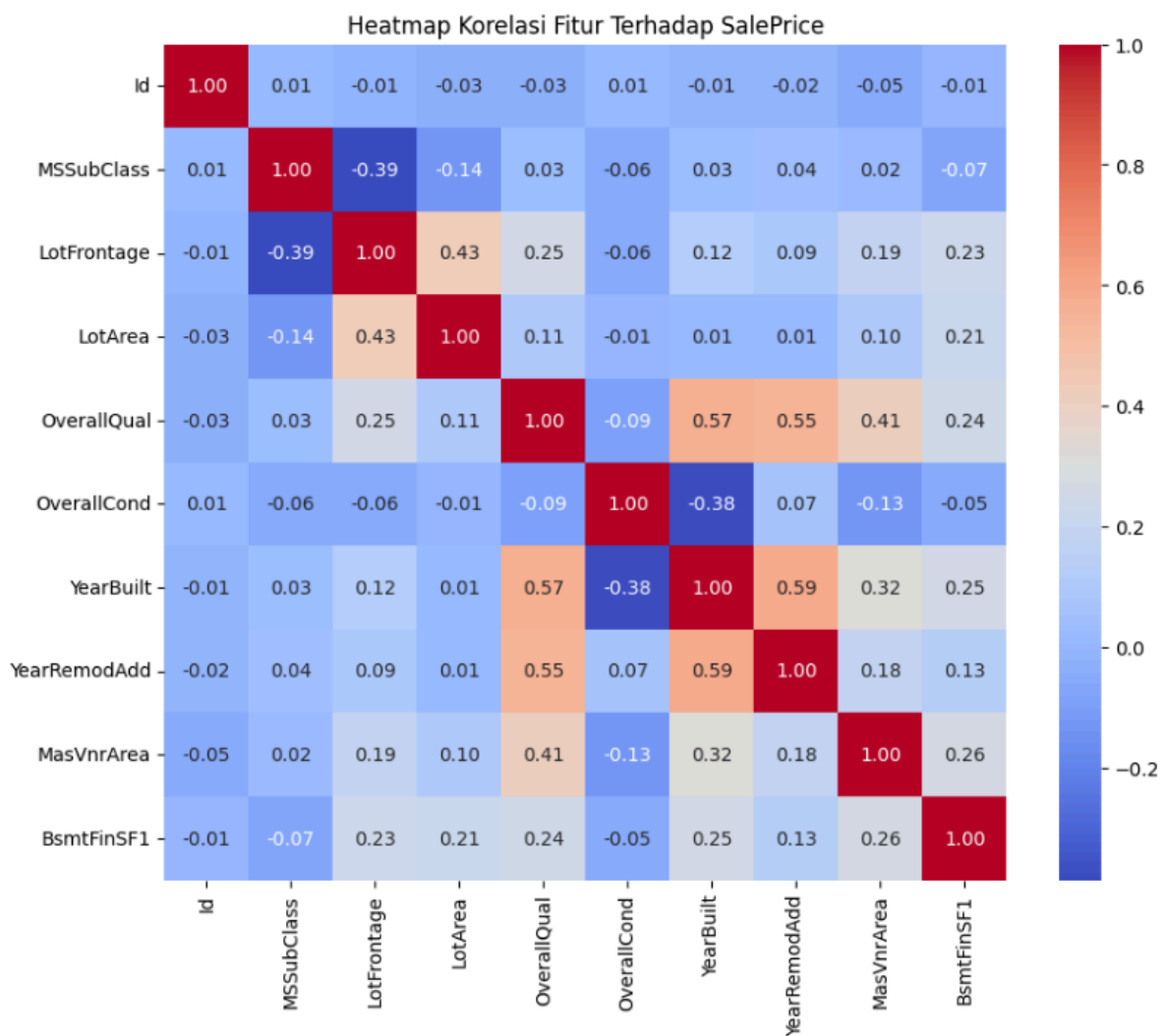
ini membantu mengidentifikasi fitur-fitur yang paling berpengaruh terhadap target SalePrice dan menunjukkan adanya potensi multikolinearitas di antara beberapa fitur numerik.

- Heatmap Korelasi Fitur Terhadap SalePrice

code:

```
top_corr_features = correlation_matrix.head(10).index
plt.figure(figsize=(10, 8))
sns.heatmap(train[top_corr_features].corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Heatmap Korelasi Fitur Terhadap SalePrice')
plt.show()
```

Output:



Interpretasi:

Berdasarkan heatmap korelasi fitur terhadap SalePrice, terlihat bahwa beberapa variabel memiliki hubungan yang cukup kuat terhadap harga penjualan. Fitur OverallQual menunjukkan korelasi paling tinggi dan positif dengan SalePrice, yang artinya semakin tinggi kualitas keseluruhan sebuah rumah, semakin tinggi pula harga jualnya. Selain itu, variabel YearBuilt dan YearRemodAdd juga menunjukkan korelasi positif yang cukup signifikan, menandakan bahwa rumah yang lebih baru atau telah mengalami renovasi cenderung memiliki harga yang lebih tinggi. MasVnrArea (luas veneer batu) dan BsmtFinSF1 (luas basement yang sudah selesai) juga memperlihatkan korelasi positif, yang berarti ukuran kedua fitur ini turut memengaruhi nilai jual rumah. Sebaliknya, fitur seperti OverallCond dan MSSubClass menunjukkan korelasi yang lebih rendah bahkan cenderung lemah terhadap SalePrice. Dari hasil ini bisa disimpulkan bahwa faktor kualitas bangunan dan ukuran area memiliki pengaruh yang lebih besar dalam menentukan harga jual rumah dibandingkan fitur lainnya.

- Mengevaluasi **hubungan antara fitur-fitur tertentu dengan SalePrice** (harga rumah) melalui visualisasi boxplot.

```
1. Boxplot: OverallQual vs SalePrice

plt.figure(figsize=(10, 6))

sns.boxplot(x='OverallQual', y='SalePrice', data=train)

plt.title('Sale Price vs Overall Quality')

plt.xlabel('Overall Quality')

plt.ylabel('Sale Price')

plt.grid(True)

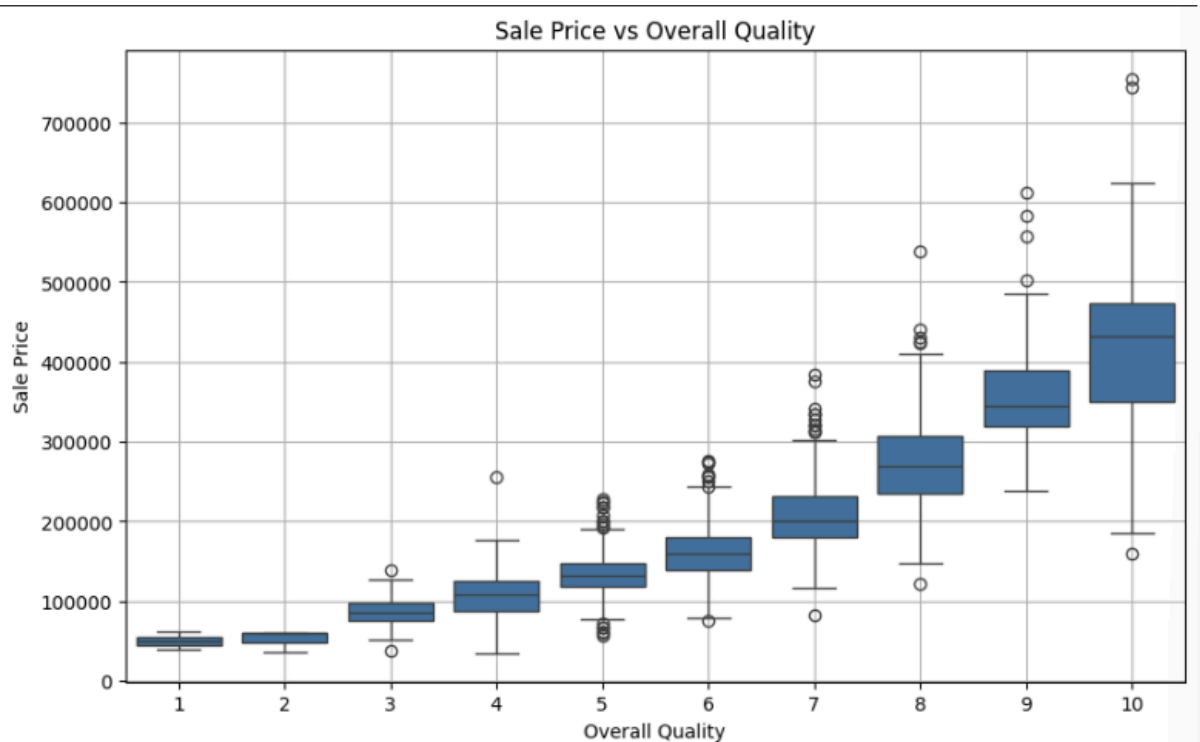
plt.show()
```

Interpretasi

Kode tersebut membuat sebuah grafik boxplot untuk menunjukkan hubungan antara kualitas keseluruhan rumah (OverallQual) dan harga jual (SalePrice). Grafik ini diambil dari data

train dan digunakan untuk melihat bagaimana harga rumah berubah berdasarkan tingkat kualitasnya. Dalam grafik ini, OverallQual diletakkan di sumbu horizontal, sementara SalePrice ada di sumbu vertikal. Hasilnya menunjukkan bahwa semakin tinggi kualitas rumah, maka semakin tinggi pula harga jualnya. Setiap kotak dalam grafik mewakili kisaran harga pada satu tingkat kualitas, dengan garis di dalam kotak menunjukkan harga tengah (median). Selain itu, titik-titik di luar kotak menunjukkan harga yang sangat tinggi atau sangat rendah dibandingkan harga lain dalam kelompok tersebut, yang disebut outlier. Kesimpulannya, grafik ini memperlihatkan bahwa kualitas rumah sangat berpengaruh terhadap harga jualnya, dan OverallQual merupakan salah satu fitur penting dalam analisis harga rumah.

Output



2. Boxplot: Neighborhood vs SalePrice

```
plt.figure(figsize=(14, 6))

sns.boxplot(x='Neighborhood', y='SalePrice', data=train)

plt.title('Sale Price vs Neighborhood')

plt.xlabel('Neighborhood')
```



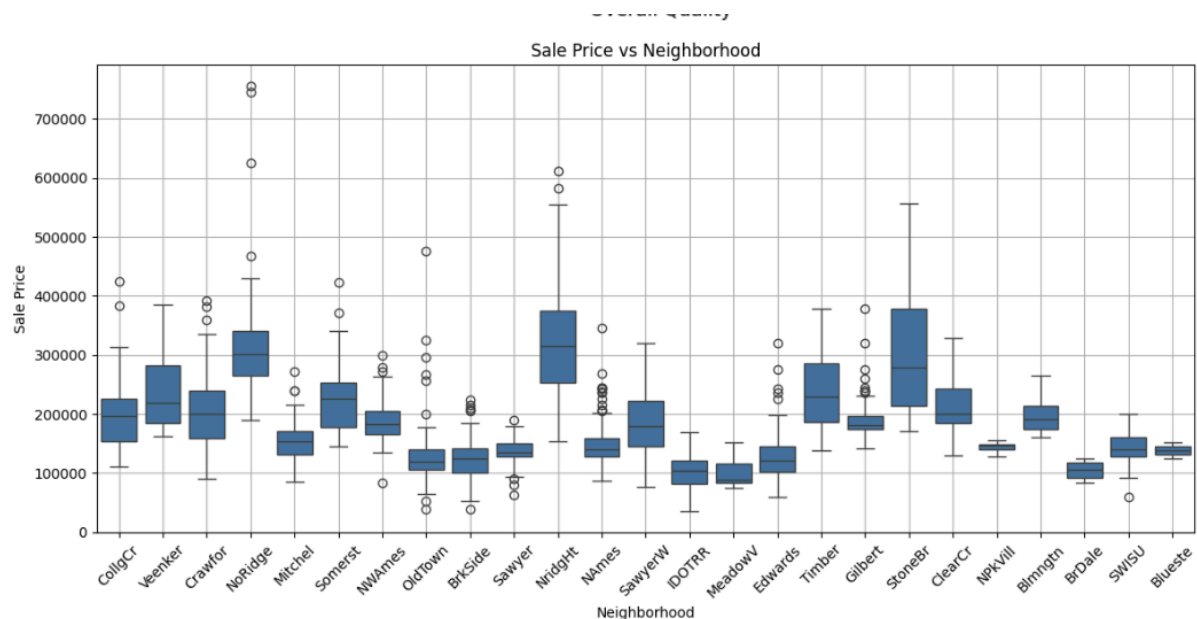
```
plt.ylabel('Sale Price')

plt.xticks(rotation=45)

plt.grid(True)

plt.show()
```

Output



Interpretasi

grafik boxplot yang menunjukkan hubungan antara lingkungan tempat rumah berada (Neighborhood) dengan harga jual rumah (SalePrice). Grafik ini membantu kita melihat perbedaan harga rumah di berbagai lingkungan.

Dari grafik ini, kita bisa lihat bahwa beberapa lingkungan seperti NoRidge, NridgHt, dan StoneBr cenderung punya harga rumah yang lebih tinggi. Sementara lingkungan seperti MeadowV, IDOTRR, dan BrDale cenderung punya harga rumah yang lebih rendah. Titik-titik di luar kotak menunjukkan harga yang tidak biasa (sangat mahal atau sangat murah) untuk lingkungan tersebut.

3. Boxplot: GarageCars vs SalePrice

```
plt.figure(figsize=(10, 6))

sns.boxplot(x='GarageCars', y='SalePrice', data=train)

plt.title('Sale Price vs Number of Garage Cars')

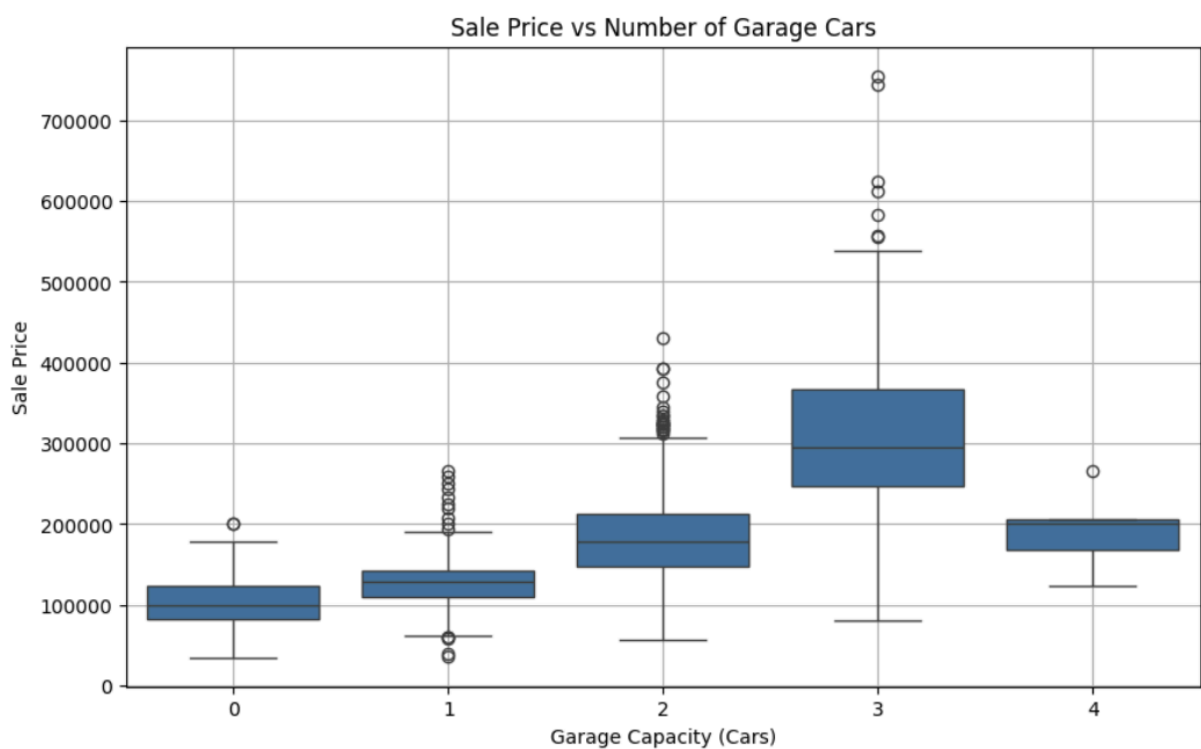
plt.xlabel('Garage Capacity (Cars)')

plt.ylabel('Sale Price')

plt.grid(True)

plt.show()
```

Output



Interpretasi

Hasil grafik memperlihatkan bahwa semakin besar kapasitas garasi, umumnya harga rumah juga semakin mahal. Misalnya, rumah dengan garasi untuk 2 atau 3 mobil cenderung memiliki harga jual yang lebih tinggi dibanding rumah yang hanya muat 1 mobil atau bahkan tidak punya garasi sama sekali.

Garis di tengah kotak menunjukkan harga rata-rata tengah (median), dan kotak menunjukkan kisaran harga untuk setiap kapasitas garasi. Titik-titik di luar kotak adalah harga yang tidak biasa (bisa jauh lebih tinggi atau lebih rendah dari yang lain di kelompok yang sama).

```
4. Boxplot: YearBuilt (dibuat kategori) vs SalePrice

train['YearBuiltBin'] = pd.cut(train['YearBuilt'],
                                bins=[1870, 1940, 1970, 2000, 2010],
                                labels=["<1940", "1940-1970",
"1970-2000", "2000+"])

plt.figure(figsize=(10, 6))

sns.boxplot(x='YearBuiltBin', y='SalePrice', data=train)

plt.title('Sale Price vs Year Built Category')

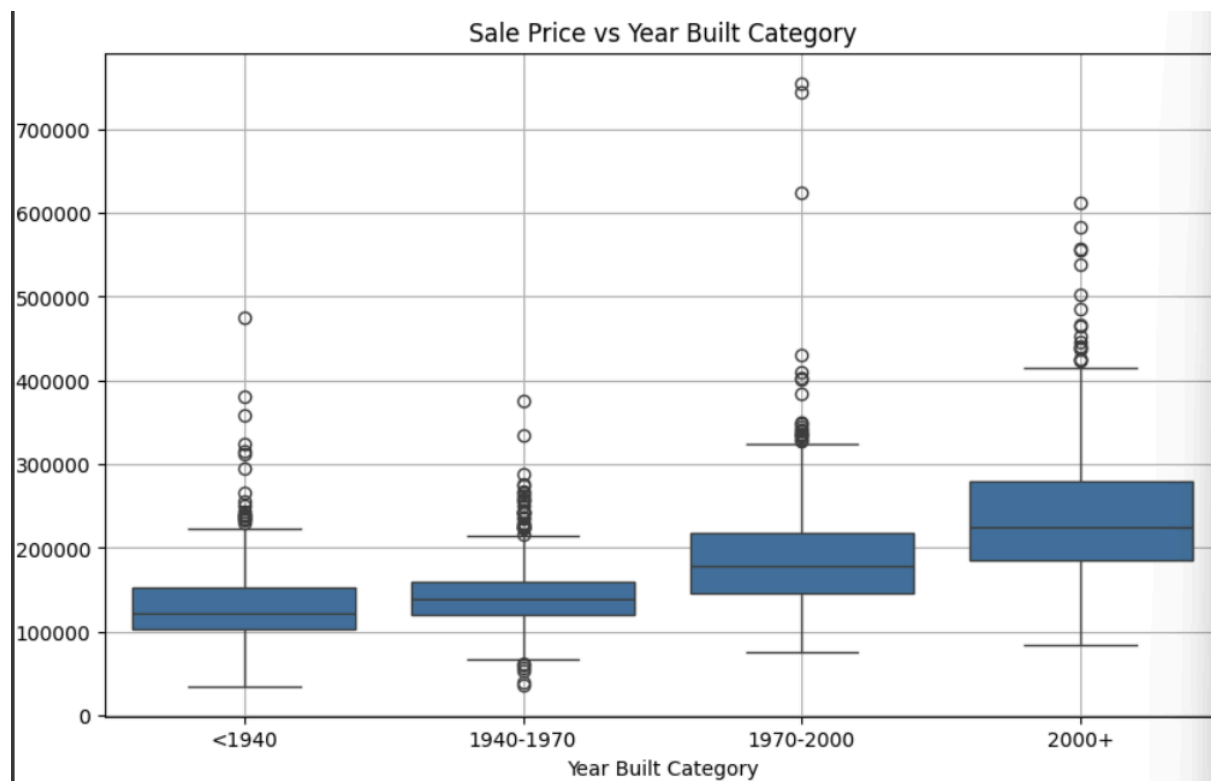
plt.xlabel('Year Built Category')

plt.ylabel('Sale Price')

plt.grid(True)

plt.show()
```

Output



Interpretasi

Kode ini digunakan untuk mengelompokkan rumah berdasarkan tahun dibangunnya menjadi beberapa kategori, yaitu sebelum 1940, antara 1940-1970, antara 1970-2000, dan tahun 2000 ke atas. Kategori ini dimasukkan ke dalam kolom baru bernama YearBuiltBin. Setelah itu, dibuat sebuah grafik boxplot untuk memperlihatkan hubungan antara kategori tahun dibangun dengan harga jual rumah (SalePrice). Dari hasil grafik, terlihat bahwa semakin baru rumah dibangun, semakin tinggi pula harga jualnya. Rumah yang dibangun setelah tahun 2000 memiliki harga jual tertinggi, sementara rumah yang dibangun sebelum tahun 1940 cenderung memiliki harga yang lebih rendah. Grafik ini juga menunjukkan sebaran harga untuk tiap kelompok tahun, termasuk harga rata-rata dan outlier (harga yang sangat tinggi atau rendah). Kesimpulannya, tahun dibangunnya rumah berpengaruh besar terhadap harga jual, di mana rumah yang lebih baru biasanya lebih mahal karena kondisi yang masih bagus, desain yang lebih modern, dan teknologi bangunan yang lebih baru.

2.3 Validation Data

Agar model dapat dievaluasi secara objektif, data dibagi menjadi dua bagian:

- Training set (80%): Digunakan untuk membangun dan melatih model.
- Testing set (20%):
- Digunakan untuk menguji generalisasi model terhadap data baru.

```
from sklearn.model_selection import train_test_split

# Split data untuk training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f"Data telah dibagi menjadi:\n- X_train: {X_train.shape}\n- X_test: {X_test.shape}")

Data telah dibagi menjadi:
- X_train: (1168, 246)
- X_test: (292, 246)
```

Gambar 3 jumlah Training set (80%): digunakan untuk membangun dan melatih model.

Testing set (20%)

Selain itu, digunakan teknik:

- Cross-validation (5-Fold): Memecah data training menjadi lima bagian untuk memastikan model tidak overfitting.
- Holdout validation: Menguji performa akhir pada data testing yang belum pernah dilihat model sebelumnya.

Proses validasi ini sangat penting untuk memastikan bahwa model yang dikembangkan tidak hanya menghafal data training, tetapi mampu memprediksi dengan baik untuk data real-world yang sebelumnya tidak dikenal.

BAB 3 DATA PREPARATION

3.1 Data Selection

Proses data selection dilakukan untuk memilih fitur-fitur yang relevan dan berkorelasi tinggi terhadap target variabel yaitu SalePrice. Dari hasil analisis sebelumnya seperti heatmap korelasi dan eksplorasi deskriptif, fitur-fitur berikut diprioritaskan karena memiliki kontribusi besar dalam memengaruhi harga rumah:

- OverallQual (Kualitas keseluruhan material dan finishing rumah)
- GrLivArea (Luas ruang tinggal di atas tanah)
- GarageCars (Kapasitas garasi berdasarkan jumlah mobil)
- GarageArea (Luas area garasi)
- TotalBsmtSF (Total luas basement)
- 1stFlrSF (Luas lantai satu)
- YearBuilt (Tahun pembangunan)
- Neighborhood (Lingkungan perumahan)
- YearRemodAdd (Tahun renovasi terakhir)

Selain itu, beberapa fitur yang memiliki banyak missing values seperti PoolQC, MiscFeature, dan Alley dipertimbangkan untuk dihapus atau diimputasi tergantung pada dampaknya dalam analisis lebih lanjut.

3.2 Data Cleaning

Tahapan ini fokus untuk mengatasi data yang hilang (missing values), duplikasi, dan nilai outlier yang berpotensi menurunkan performa model. Adapun langkah-langkah pembersihan data yang dilakukan adalah sebagai berikut:

- Handling Missing Values:
 - Fitur yang bersifat kategorikal seperti Alley, Fence, MiscFeature, PoolQC diisi dengan string 'None' karena menunjukkan ketidakadaan fasilitas tersebut.
 - Fitur numerik seperti LotFrontage diisi dengan nilai median dari masing-masing lingkungan (Neighborhood) untuk menjaga distribusi data tetap representatif.
 - GarageYrBlt yang bernilai kosong diasumsikan tidak memiliki garasi, sehingga diisi dengan 0.

- MasVnrArea yang kosong diisi dengan 0, dan MasVnrType yang kosong diisi dengan 'None'.
- Fitur Electrical yang memiliki satu nilai kosong diisi dengan mode (nilai yang paling sering muncul) yaitu 'SBrkr'.
- Outlier Removal:
 - Deteksi outlier dilakukan dengan visualisasi boxplot untuk fitur GrLivArea, SalePrice, dan TotalBsmtSF.
 - Rumah dengan GrLivArea > 4500 dan SalePrice < 300000 dihapus karena terindikasi sebagai outlier yang tidak sesuai pola umum.
- Inconsistent Data Fixing:
 - Pemeriksaan logika antar fitur dilakukan, seperti memastikan rumah dengan GarageArea == 0 memiliki GarageType == 'None'.
 - Data yang tidak konsisten diperbaiki atau dihapus untuk mencegah error pada proses modelling.

3.3 Data Construct

Pada tahap konstruksi data, beberapa transformasi fitur dilakukan untuk memperkuat representasi data dan memudahkan proses pelatihan model:

- Feature Transformation:

Transformasi logaritmik diterapkan pada variabel SalePrice dan GrLivArea untuk mengurangi skewness:

```
train['SalePrice'] = np.log1p(train['SalePrice'])
```

```
train['GrLivArea'] = np.log1p(train['GrLivArea'])
```

- Hal ini dilakukan karena distribusi awal kedua variabel sangat condong ke kanan, yang dapat menghambat performa model regresi.
- Binning:
 - Fitur YearBuilt dibagi menjadi beberapa kategori (binning) seperti:
 - <1940
 - 1940-1970

- 1970-2000
- 2000+
- Binning ini bertujuan untuk membantu model mengenali pengaruh usia bangunan dalam bentuk kategori, yang seringkali lebih bermakna dibandingkan nilai absolutnya.
- Encoding:
 - Fitur kategorikal dikonversi menjadi representasi numerik menggunakan:
 - Label Encoding untuk fitur ordinal seperti ExterQual dan BsmtQual.
 - One-Hot Encoding untuk fitur nominal seperti Neighborhood dan HouseStyle agar tidak memperkenalkan urutan yang tidak ada.

3.4 Labeling Data

Proses labeling dalam proyek ini merujuk pada penetapan SalePrice sebagai target variabel dalam supervised learning. Karena SalePrice memiliki distribusi yang skewed, dilakukan transformasi logaritmik untuk memastikan data label lebih normal dan model lebih mudah belajar:

- Label akhir:


```
y = np.log1p(train['SalePrice'])
```
- Hal ini juga mencegah prediksi yang bias terhadap harga rumah di kelas ekstrem (terlalu tinggi atau terlalu rendah).

3.5 Data Integration

Integrasi data dilakukan dengan cara menggabungkan dataset training dan testing selama proses pembersihan dan transformasi fitur agar konsistensi preprocessing tetap terjaga di kedua bagian data. Langkah-langkahnya meliputi:

- Merging Train dan Test Data:


```
all_data = pd.concat([train.drop(['SalePrice'], axis=1), test], sort=False)
```
- Transformasi Fitur Secara Seragam:
 - Semua fitur yang telah di-encode dan dibersihkan diterapkan baik ke data train maupun test.
 - Hal ini mencegah data leakage dan memastikan distribusi fitur seragam pada saat proses deployment nanti.

- Splitting Kembali: Setelah proses preprocessing selesai, dataset dipisahkan kembali menjadi:
 - X_train — data training untuk pelatihan model.
 - X_test — data testing untuk proses prediksi.

Dengan proses data preparation ini, diharapkan data sudah dalam kondisi optimal, siap digunakan untuk proses pemodelan dan evaluasi lebih lanjut.

BAB 4 MODELLING

4.1 Build Model

BAB 5 EVALUATION

BAB 6 DEPLOYMENT

DAFTAR PUSTAKA