

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
2.2 Describe Data.....	6
2.3 Validation Data	25
BAB 3 DATA PREPARATION.....	27
3.1 Data Selection	28
3.2 Data Cleaning	28
3.3 Data Construct	31
3.4 Labeling Data.....	32
3.5 Data Integration.....	33
BAB 4 MODELLING.....	34
4.1 Build Model.....	34
BAB 5 EVALUATION.....	35
BAB 6 DEPLOYMENT	36
DAFTAR PUSTAKA	37

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.

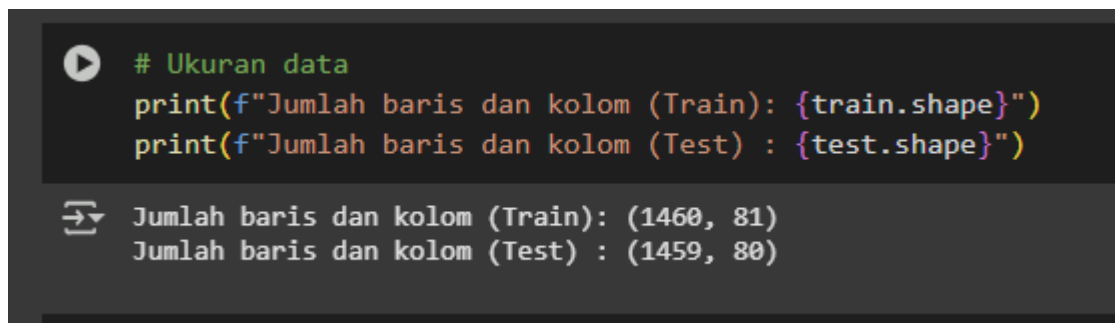
BAB 2 DATA UNDERSTANDING

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	MSClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	RestroomArea	BsmtFinSF1	...	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea	MiscVal	MSZoning	YrSold	SalePrice
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	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	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.685262	443.639726	...	94.244521	46.660274	21.954110	3.409589	15.060959	2.758904	43.489041	6.321918	2007.815753	180921.195890
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.066207	456.088091	...	125.338794	66.256028	61.119149	29.317331	55.757415	40.177307	496.123024	2.703626	1.328895	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	2094.000000	166.000000	712.250000	...	168.000000	68.000000	0.000000	0.000000	0.000000	0.000000	0.000000	8.000000	2008.000000	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

0 rows x 38 columns

Interpretasi:

menunjukkan ringkasan statistik dari 38 kolom numerik dalam dataset. Informasi ini mencakup jumlah data (count), nilai rata-rata (mean), standar deviasi (std), nilai terkecil (min), nilai kuartil 1 (25%), median atau kuartil 2 (50%), kuartil 3 (75%), dan nilai terbesar (max). Sebagian besar kolom memiliki jumlah data yang lengkap (1.460 baris), tetapi ada beberapa kolom seperti LotFrontage dan MasVnrArea yang memiliki jumlah data lebih sedikit, yang berarti terdapat data yang kosong (missing).

Harga rumah (SalePrice) memiliki nilai rata-rata sekitar 180.921 dengan harga tertinggi mencapai 755.000 dan terendah 34.900. Ini menunjukkan bahwa harga rumah di dataset sangat bervariasi. Nilai standar deviasi yang cukup besar (± 79.442) juga memperkuat bahwa penyebaran harga cukup lebar. Beberapa fitur seperti LotArea, BsmtFinSF1, dan GarageArea menunjukkan adanya perbedaan besar antara nilai maksimum dan nilai kuartil atas, menandakan adanya rumah-rumah yang memiliki ukuran atau fitur jauh lebih besar dibandingkan yang lain (outlier).

Sementara itu, banyak fitur seperti PoolArea, ScreenPorch, dan 3SsnPorch memiliki nilai tengah (median) sebesar nol. Artinya, sebagian besar rumah tidak memiliki fasilitas tersebut, namun ada beberapa rumah yang memilikinya dengan ukuran yang cukup besar.

Fitur penilaian kondisi rumah seperti OverallQual (kualitas keseluruhan) dan OverallCond (kondisi keseluruhan) memiliki nilai antara 1 sampai 10. Rata-rata nilainya adalah sekitar 6, yang menunjukkan bahwa sebagian besar rumah berada dalam kondisi cukup baik.

- 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.

- Visualisasi Distribusi Sale Price (target):

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

Code:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Set style
sns.set(style="whitegrid")

# Hitung statistik dasar
mean_price = train['SalePrice'].mean()
median_price = train['SalePrice'].median()
skewness = train['SalePrice'].skew()
kurtosis = train['SalePrice'].kurt()

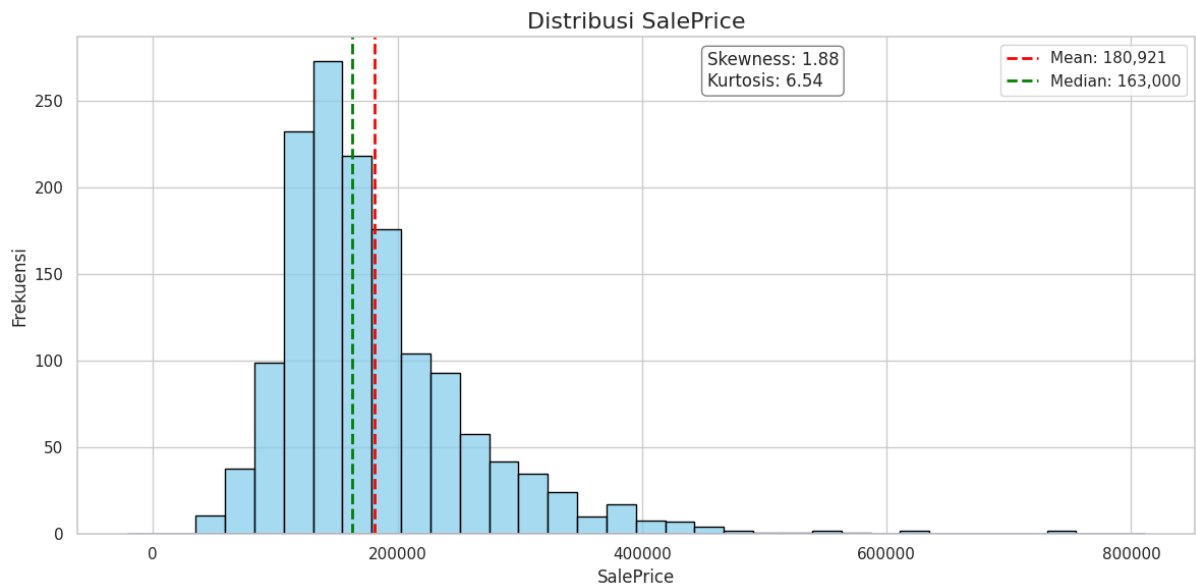
# Plot
plt.figure(figsize=(12, 6))
sns.histplot(train['SalePrice'], kde=False, bins=30, color='skyblue', edgecolor='black')
sns.kdeplot(train['SalePrice'], color='darkblue', linewidth=2)

# Tambahkan garis mean dan median
plt.axvline(mean_price, color='red', linestyle='--', linewidth=2, label=f'Mean: {mean_price:,.0f}')
plt.axvline(median_price, color='green', linestyle='--', linewidth=2, label=f'Median: {median_price:,.0f}')

# Anotasi Skewness dan Kurtosis
plt.text(x=train['SalePrice'].max()*0.6, y=plt.gca().get_ylim()[1]*0.9,
        s=f'Skewness: {skewness:.2f}\nKurtosis: {kurtosis:.2f}',
        fontsize=12, bbox=dict(boxstyle="round", fc="w", ec="gray"))

# Tambahkan estetika
plt.title("Distribusi SalePrice", fontsize=16)
plt.xlabel("SalePrice")
plt.ylabel("Frekuensi")
plt.legend()
plt.tight_layout()
plt.show()
```

Output:



Interpretasi:

Tujuan dari kode di atas adalah untuk menganalisis dan memvisualisasikan distribusi harga jual rumah (SalePrice) dalam dataset. Berikut adalah rincian tujuan kode tersebut:

1. Menghitung Statistik Dasar: Kode ini menghitung nilai mean, median, skewness, dan kurtosis dari SalePrice untuk memberikan gambaran umum mengenai distribusi data dan karakteristiknya.
 - Mean memberikan nilai rata-rata dari harga jual rumah.
 - Median memberikan nilai tengah, yang berguna untuk memahami posisi tengah distribusi data.
 - Skewness mengukur ketidakseimbangan distribusi, apakah data cenderung miring ke kiri atau ke kanan.
 - Kurtosis mengukur sejauh mana distribusi data memiliki puncak yang tajam atau datar.
2. Memvisualisasikan Distribusi Data: Menggunakan histogram dan KDE (Kernel Density Estimation), kode ini menggambarkan bagaimana harga jual rumah tersebar.
 - Histogram menunjukkan frekuensi distribusi nilai harga jual rumah.
 - KDE plot memberikan gambaran halus dari distribusi tersebut untuk memudahkan pemahaman pola distribusi.

3. Menambahkan Garis Mean dan Median: Garis mean dan median ditambahkan untuk memberikan wawasan mengenai posisi harga tengah dan rata-rata dalam distribusi. Ini membantu dalam memahami apakah distribusi data simetris atau condong ke satu arah.
4. Anotasi Skewness dan Kurtosis: Menyediakan informasi tentang skewness dan kurtosis secara langsung pada grafik, yang memungkinkan pengguna untuk dengan mudah melihat karakteristik distribusi data.

- Heatmap korelasi antar fitur numerik

code:

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

# Pisahkan fitur numerik dan kategorikal
numeric_feats = train.select_dtypes(include=['int64',
'float64'])
categorical_feats = train.select_dtypes(include=['object'])

# Label Encoding untuk fitur kategorikal agar bisa dihitung
korelasinya
label_encoded = categorical_feats.copy()
le = LabelEncoder()
for col in label_encoded.columns:
    label_encoded[col] = le.fit_transform(label_encoded[col].astype(str))

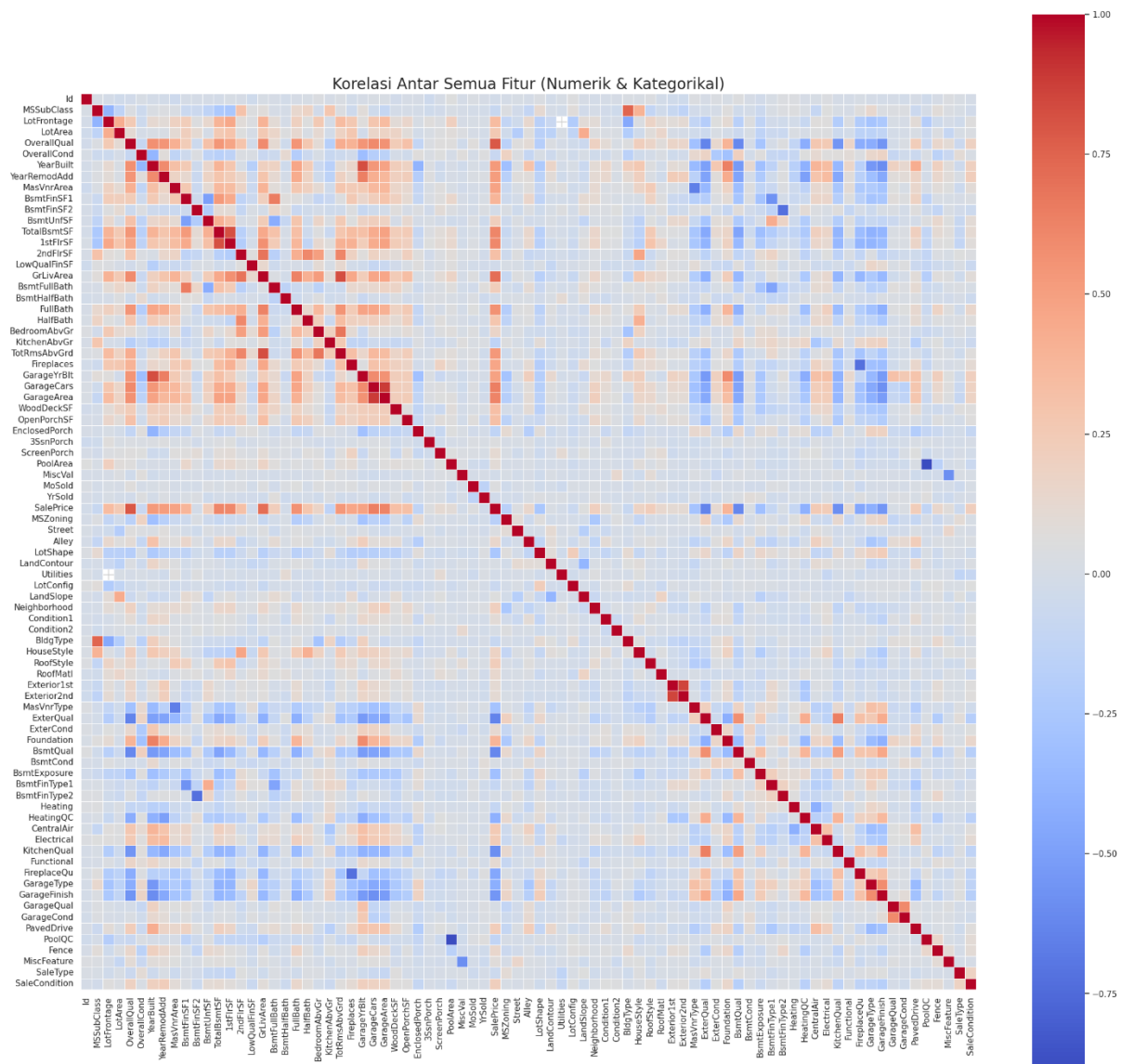
# Gabungkan fitur numerik dan kategorikal yang sudah di-encode
combined_data = pd.concat([numeric_feats, label_encoded],
axis=1)

# Hitung korelasi
corr_matrix = combined_data.corr()

# Visualisasi Heatmap Korelasi
plt.figure(figsize=(22, 20))
sns.heatmap(
    corr_matrix,
    cmap='coolwarm',
    annot=False,
    linewidths=0.5,
    cbar=True,
    square=True,
    xticklabels=True,
    yticklabels=True
)
```

```
plt.title('Korelasi Antar Semua Fitur (Numerik &
Kategorikal)', fontsize=20)
plt.xticks(rotation=90)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```

Output:



Interpretasi:

Gambar heatmap di atas menunjukkan hubungan korelasi antar seluruh fitur dalam dataset, baik fitur numerik maupun fitur kategorikal yang telah diubah ke dalam format numerik melalui proses *Label Encoding*. Korelasi yang ditampilkan dihitung menggunakan metode Pearson, yang mengukur sejauh mana hubungan linier antara dua variabel. Warna merah pada

heatmap menunjukkan korelasi positif, yaitu apabila nilai suatu variabel meningkat, maka nilai variabel lainnya juga cenderung meningkat. Sebaliknya, warna biru menunjukkan korelasi negatif, yaitu apabila nilai satu variabel meningkat, maka nilai variabel lainnya cenderung menurun. Sementara itu, warna yang mendekati putih mengindikasikan bahwa tidak terdapat hubungan linier yang kuat antara kedua variabel tersebut.

Dari hasil visualisasi ini, dapat diamati bahwa beberapa fitur memiliki korelasi yang cukup kuat satu sama lain. Misalnya, fitur seperti *OverallQual*, *GrLivArea*, dan *GarageCars* tampak memiliki hubungan positif yang cukup kuat terhadap variabel *SalePrice*. Hal ini menunjukkan bahwa rumah dengan kualitas keseluruhan yang lebih baik, luas area bangunan yang lebih besar, dan kapasitas garasi yang memadai cenderung memiliki harga jual yang lebih tinggi. Di sisi lain, terdapat juga fitur-fitur yang memiliki korelasi negatif terhadap *SalePrice*, yang menunjukkan bahwa peningkatan nilai pada fitur-fitur tersebut justru berpotensi menurunkan harga jual rumah.

Secara keseluruhan, heatmap ini sangat berguna dalam proses eksplorasi data, khususnya untuk memahami hubungan antar fitur dan membantu dalam pemilihan fitur yang relevan. Fitur-fitur dengan korelasi yang sangat tinggi satu sama lain dapat dipertimbangkan untuk direduksi guna menghindari masalah multikolinearitas dalam proses pemodelan selanjutnya. Sementara itu, fitur yang menunjukkan korelasi kuat terhadap variabel target dapat dijadikan fokus utama dalam pengembangan model prediksi harga rumah.

- Korelasi Antar Fitur

Code:

```
import pandas as pd

train_encoded = pd.get_dummies(train, drop_first=True)

correlation_matrix = train_encoded.corr()

saleprice_corr = correlation_matrix['SalePrice']

high_corr_features = saleprice_corr[saleprice_corr > 0.5].sort_values(ascending=False)

print(high_corr_features)
```

Output:

```
SalePrice      1.000000
OverallQual    0.790982
GrLivArea      0.708624
GarageCars     0.640409
GarageArea     0.623431
TotalBsmntSF   0.613581
1stFlrSF       0.605852
FullBath       0.560664
TotRmsAbvGrd   0.533723
YearBuilt      0.522897
YearRemodAdd    0.507101
Name: SalePrice, dtype: float64
```

- Visualisasi Distribusi Harga Jual Berdasarkan Fitur-Fitur Properti Menggunakan Boxplot

code:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

features = ['OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea',
            'TotalBsmntSF', '1stFlrSF', 'FullBath', 'TotRmsAbvGrd',
            'YearBuilt', 'YearRemodAdd']

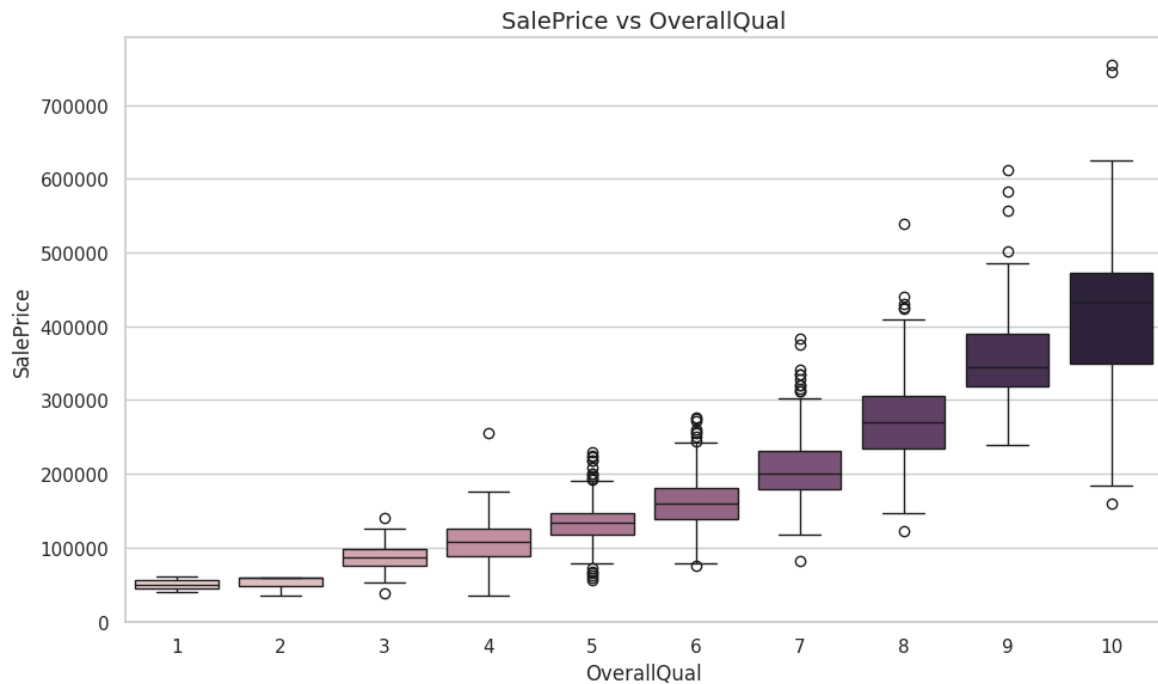
for feature in features:
    plt.figure(figsize=(10, 6))

    if train[feature].nunique() < 20:
        sns.boxplot(x=train[feature], y=train['SalePrice'], hue=train[feature], legend=False)
    else:
        binned_feature = pd.qcut(train[feature], q=4, duplicates='drop')
        sns.boxplot(x=binned_feature, y=train['SalePrice'], hue=binned_feature, legend=False)
        plt.xticks(rotation=45)

    plt.title(f'SalePrice vs {feature}', fontsize=14)
    plt.xlabel(feature)
    plt.ylabel('SalePrice')
    plt.tight_layout()
    plt.show()
```

Output:

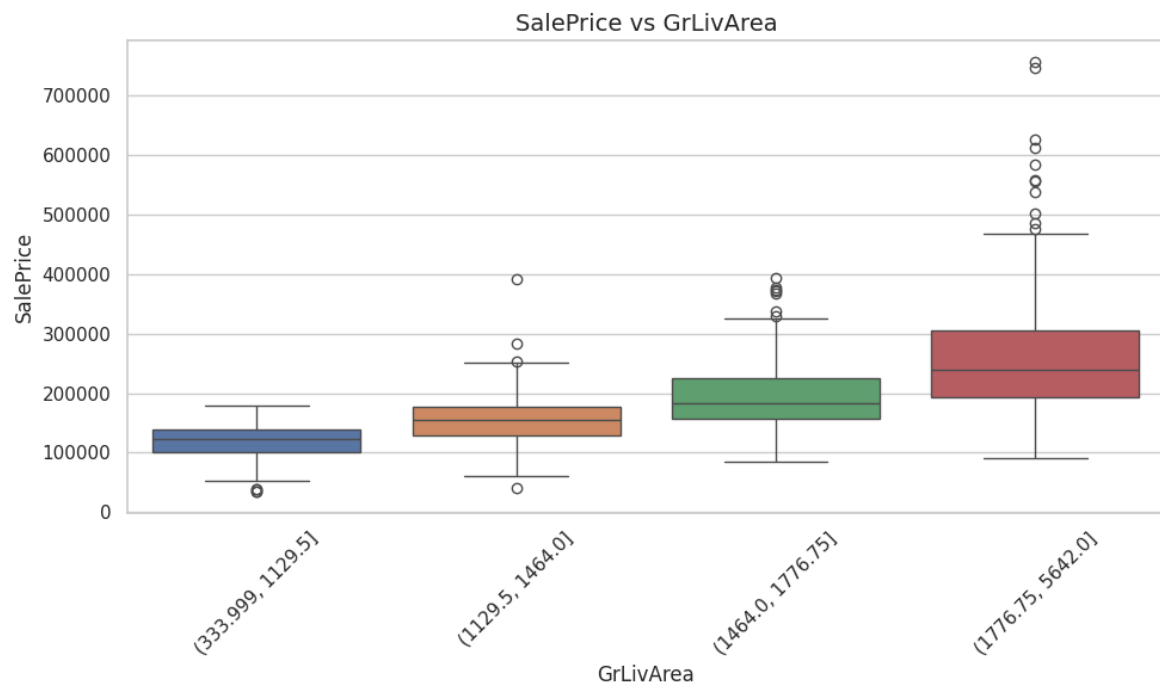
'OverallQual'



Interpretasi Boxplot:

1. **Tren Peningkatan:** Terlihat dengan jelas bahwa semakin tinggi nilai OverallQual, semakin tinggi pula median (garis tengah di boxplot) dari SalePrice. Ini menunjukkan adanya tren positif yang kuat antara kualitas material dan finishing rumah dengan harga jualnya. Rumah dengan kualitas lebih baik cenderung memiliki harga jual yang lebih tinggi.
2. **Variabilitas Harga:** Rentang interquartile (ukuran boxplot) juga cenderung meningkat seiring dengan OverallQual. Hal ini mengindikasikan bahwa variabilitas harga rumah juga meningkat seiring dengan peningkatan kualitas. Rumah dengan kualitas tinggi memiliki rentang harga yang lebih lebar dibandingkan rumah dengan kualitas rendah.
3. **Outlier:** Terdapat beberapa outlier (titik-titik di luar whisker boxplot) pada beberapa level OverallQual. Ini menunjukkan adanya beberapa rumah yang memiliki harga jual di luar rentang yang umum untuk kualitas tersebut. Outlier ini bisa jadi merupakan data yang perlu diinvestigasi lebih lanjut.
4. **Distribusi Harga:** Distribusi SalePrice untuk setiap level OverallQual cenderung miring ke kanan (right-skewed), artinya lebih banyak rumah dengan harga di bawah median daripada di atas median. Ini adalah pola yang umum pada data harga rumah.

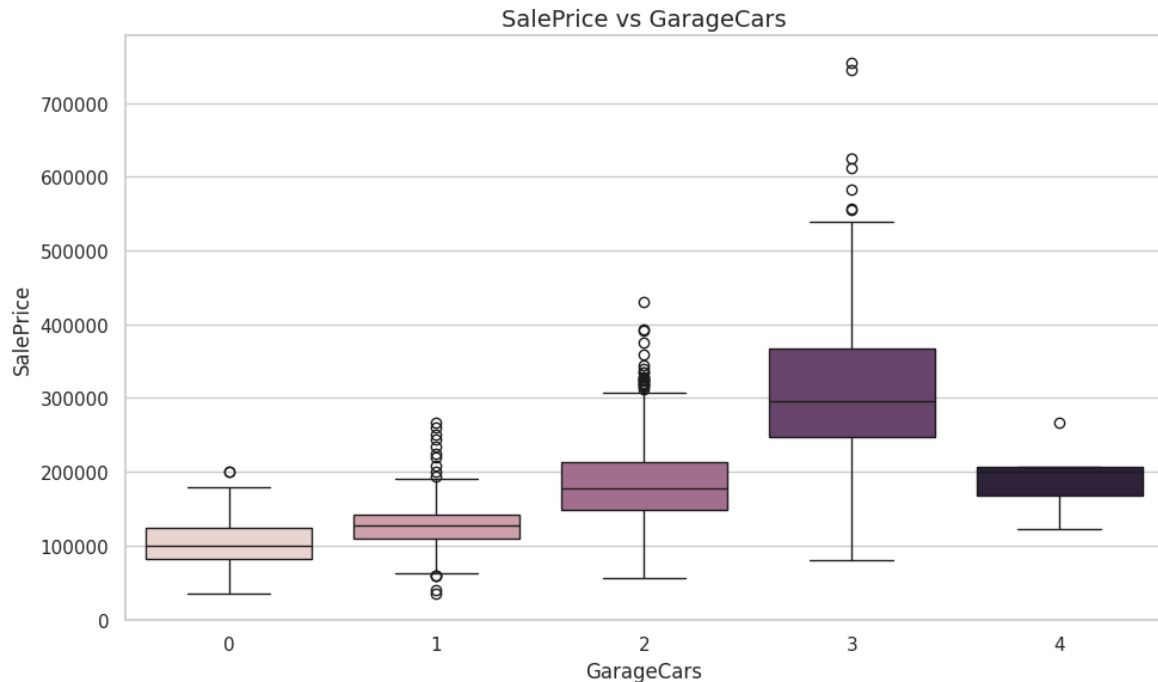
'GrLivArea',



Interpretasi:

Boxplot "SalePrice vs GrLivArea" menunjukkan bahwa terdapat hubungan positif antara luas area hunian di atas permukaan tanah (GrLivArea) dengan harga jual rumah (SalePrice). Seiring bertambahnya GrLivArea, median SalePrice juga cenderung meningkat, mengindikasikan bahwa rumah dengan luas area hunian yang lebih besar umumnya memiliki harga jual yang lebih tinggi. Namun, peningkatan GrLivArea juga diiringi dengan peningkatan variabilitas harga, terlihat dari rentang interquartile (ukuran boxplot) yang semakin lebar. Hal ini berarti rumah dengan luas area hunian yang lebih besar memiliki rentang harga yang lebih beragam. Terdapat beberapa outlier, terutama pada GrLivArea yang lebih besar, menunjukkan adanya rumah dengan harga jual di luar rentang yang umum untuk luas area hunian tersebut. Outlier ini bisa jadi merupakan rumah dengan kondisi atau fitur khusus yang membuatnya lebih mahal atau lebih murah, atau mungkin juga kesalahan data. Secara keseluruhan, boxplot ini menggambarkan bahwa GrLivArea merupakan faktor penting yang mempengaruhi SalePrice, meskipun terdapat variasi dan outlier yang perlu diperhatikan lebih lanjut.

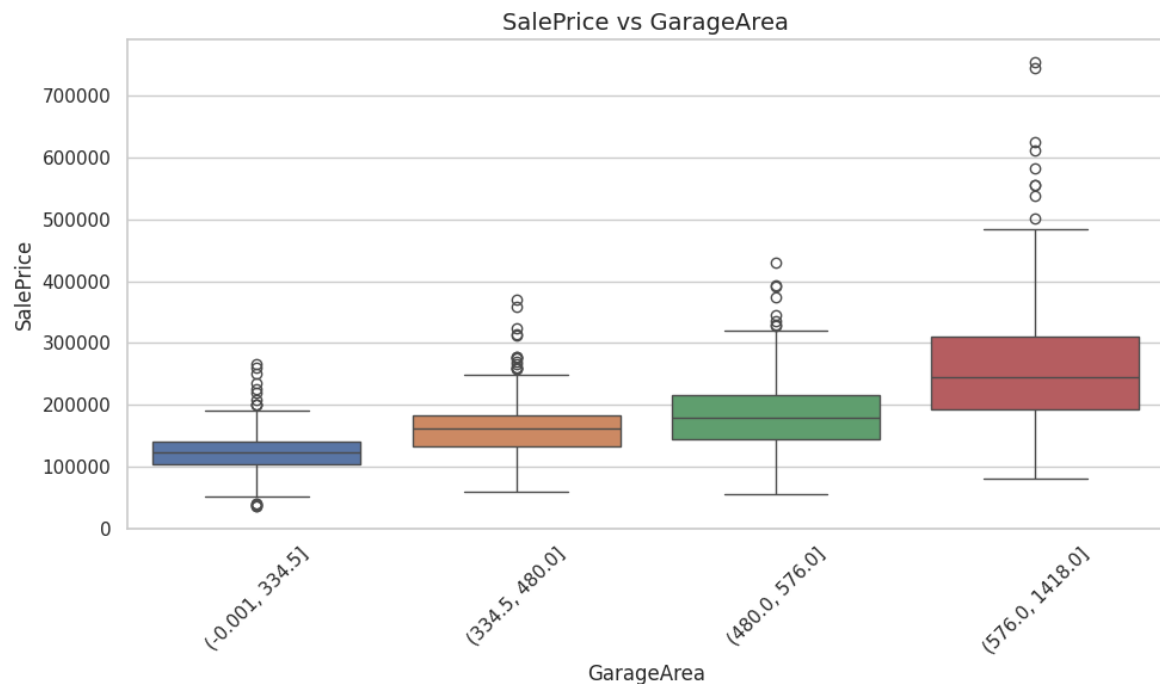
'GarageCars'



Interpretasi:

Boxplot "SalePrice vs GarageCars" menunjukkan hubungan yang cukup jelas antara jumlah kapasitas mobil di garasi (GarageCars) dengan harga jual rumah (SalePrice). Secara umum, semakin banyak kapasitas mobil yang dapat ditampung di garasi, semakin tinggi pula median SalePrice. Rumah tanpa garasi (GarageCars = 0) memiliki harga jual terendah. Kemudian, harga jual meningkat secara signifikan untuk rumah dengan kapasitas garasi 1, 2, dan 3 mobil. Namun, peningkatan harga jual relatif kecil atau bahkan stagnan ketika kapasitas garasi mencapai 4 mobil. Hal ini mengindikasikan bahwa kapasitas garasi hingga 3 mobil merupakan faktor yang cukup penting dalam menentukan harga jual rumah, tetapi kapasitas lebih dari itu mungkin tidak memberikan pengaruh yang signifikan. Terdapat sedikit outlier pada kategori GarageCars = 3 dan 4, yang menunjukkan adanya beberapa rumah dengan harga jual di luar rentang umum untuk kapasitas garasi tersebut. Secara keseluruhan, boxplot ini menggambarkan bahwa GarageCars merupakan faktor yang cukup penting

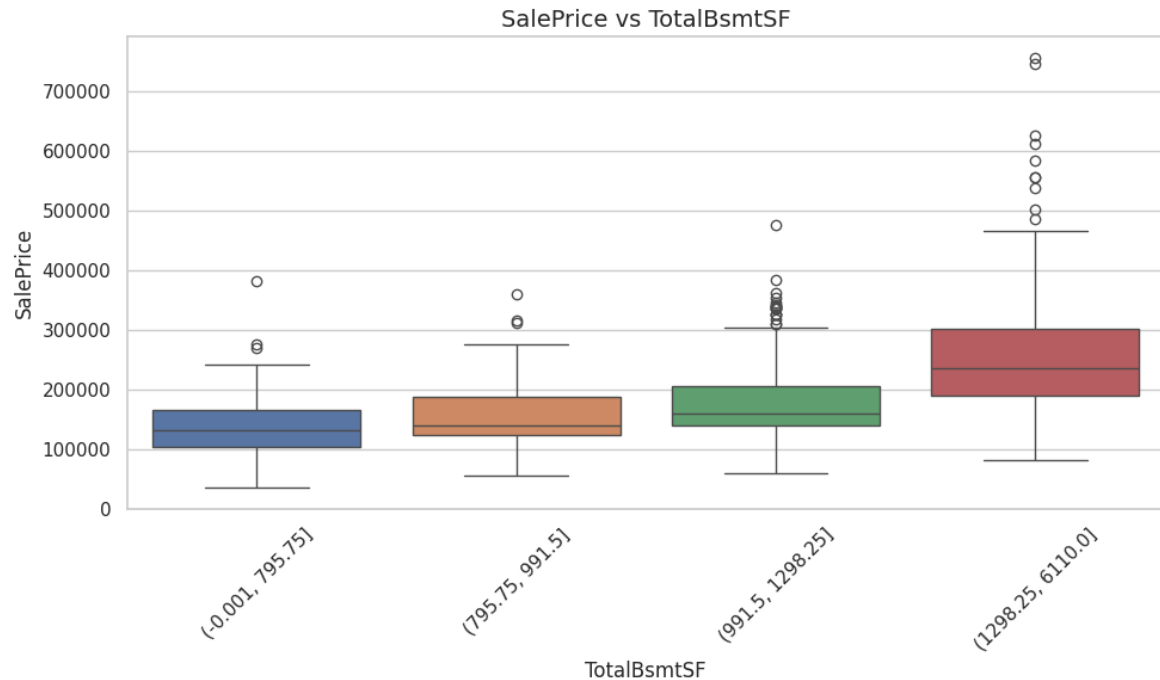
'GarageArea'



Interpretasi:

Boxplot "SalePrice vs GarageArea" menunjukkan hubungan positif antara luas garasi (GarageArea) dengan harga jual rumah (SalePrice). Seiring bertambahnya luas garasi, median SalePrice juga cenderung meningkat, mengindikasikan bahwa rumah dengan garasi yang lebih luas umumnya memiliki harga jual yang lebih tinggi. Namun, hubungan ini tidak sejelas dan sekuat pada boxplot "SalePrice vs GarageCars". Terdapat variabilitas harga yang cukup besar pada setiap rentang luas garasi, terlihat dari rentang interquartile (ukuran boxplot) yang lebar. Hal ini menunjukkan bahwa selain luas garasi, ada faktor lain yang juga mempengaruhi harga jual rumah. Terdapat beberapa outlier, terutama pada GarageArea yang lebih besar, menunjukkan adanya rumah dengan harga jual di luar rentang yang umum untuk luas garasi tersebut. Outlier ini bisa jadi merupakan rumah dengan kondisi atau fitur khusus pada garasinya, atau mungkin juga kesalahan data. Secara keseluruhan, boxplot ini menggambarkan bahwa GarageArea merupakan faktor yang cukup berpengaruh terhadap SalePrice, meskipun hubungannya tidak sekuat dan sejelas faktor jumlah kapasitas mobil di garasi (GarageCars).

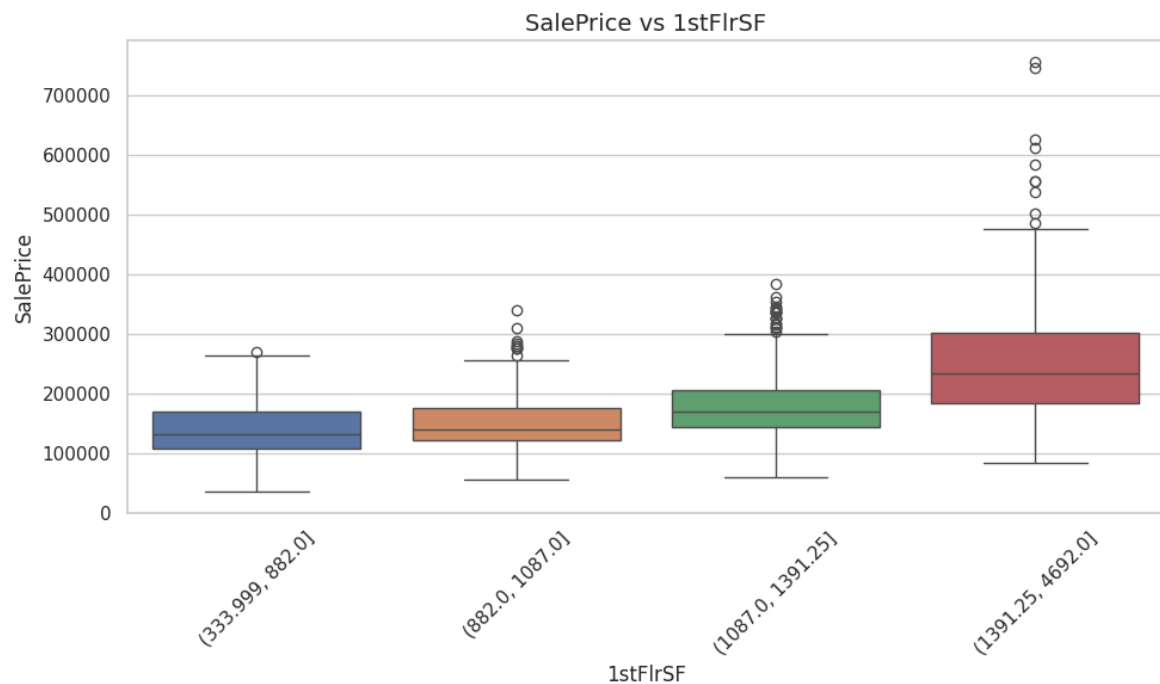
'TotalBsmtSF'



Interpretasi:

Boxplot "SalePrice vs TotalBsmtSF" menunjukkan hubungan positif antara luas total ruang bawah tanah (TotalBsmtSF) dengan harga jual rumah (SalePrice). Seiring bertambahnya luas total ruang bawah tanah, median SalePrice juga cenderung meningkat, mengindikasikan bahwa rumah dengan ruang bawah tanah yang lebih luas umumnya memiliki harga jual yang lebih tinggi. Namun, hubungan ini terlihat lebih kuat pada rentang TotalBsmtSF yang lebih rendah. Pada rentang TotalBsmtSF yang lebih tinggi, peningkatan median SalePrice cenderung melambat dan variabilitas harga semakin besar, terlihat dari rentang interquartile (ukuran boxplot) yang semakin lebar. Hal ini menunjukkan bahwa luas ruang bawah tanah yang sangat besar mungkin tidak selalu menjamin peningkatan harga jual yang signifikan. Terdapat beberapa outlier, terutama pada TotalBsmtSF yang lebih besar, menunjukkan adanya rumah dengan harga jual di luar rentang yang umum untuk luas ruang bawah tanah tersebut. Outlier ini bisa jadi merupakan rumah dengan kondisi atau fitur khusus pada ruang bawah tanahnya, atau mungkin juga kesalahan data. Secara keseluruhan, boxplot ini menggambarkan bahwa TotalBsmtSF merupakan faktor yang cukup berpengaruh terhadap SalePrice, terutama untuk luas ruang bawah tanah pada rentang menengah.

'1stFlrSF'



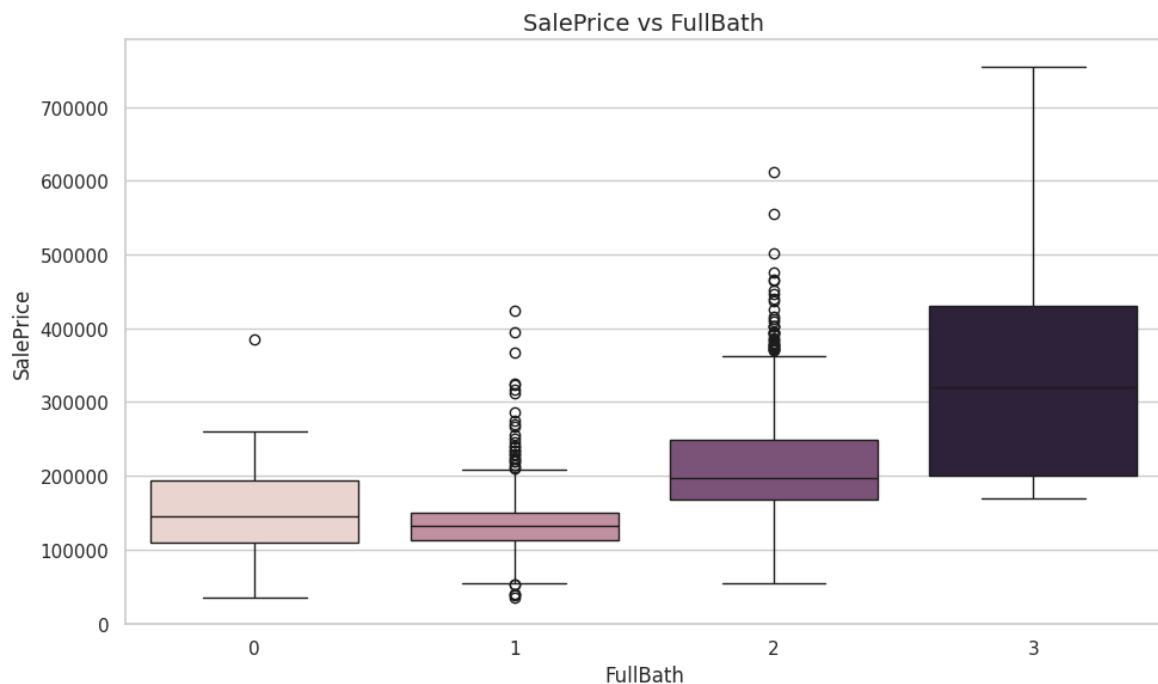
Interpretasi:

Boxplot di atas menunjukkan hubungan antara luas lantai satu rumah (1stFlrSF) yang telah dibagi dalam empat interval, dengan harga jual rumah (SalePrice). Terlihat adanya pola kenaikan harga jual seiring dengan bertambahnya luas lantai satu. Pada interval terendah, yaitu antara 333.99 hingga 882.01 kaki persegi, median harga rumah berada di kisaran 130.000, dengan sebaran harga yang relatif sempit. Interval berikutnya, yaitu 882.0 hingga 1087.0 kaki persegi, tidak menunjukkan peningkatan signifikan dalam median harga, namun tetap mencerminkan variasi harga yang sedikit lebih tinggi.

Kenaikan lebih nyata terjadi pada kelompok ketiga (1087.0 hingga 1391.25), di mana median harga meningkat mendekati 170.000–180.000, dan rentang harga menjadi lebih lebar. Sementara itu, pada interval tertinggi (1391.25 hingga 4602.0 kaki persegi), terlihat lonjakan yang signifikan baik dalam median harga (sekitar 230.000–250.000) maupun maksimum harga jual, bahkan melebihi 700.000, dengan jumlah outlier yang cukup banyak.

Interpretasi dari boxplot ini menunjukkan bahwa luas lantai pertama merupakan salah satu faktor penting yang berkontribusi terhadap peningkatan harga rumah. Rumah dengan lantai satu yang lebih luas cenderung memiliki harga jual yang lebih tinggi. Namun, kehadiran sejumlah outlier di setiap kategori juga menandakan bahwa luas lantai bukan satu-satunya penentu harga, karena faktor lain seperti kualitas bangunan, lokasi, dan fasilitas tambahan juga turut memengaruhi nilai rumah secara keseluruhan.

'FullBath'



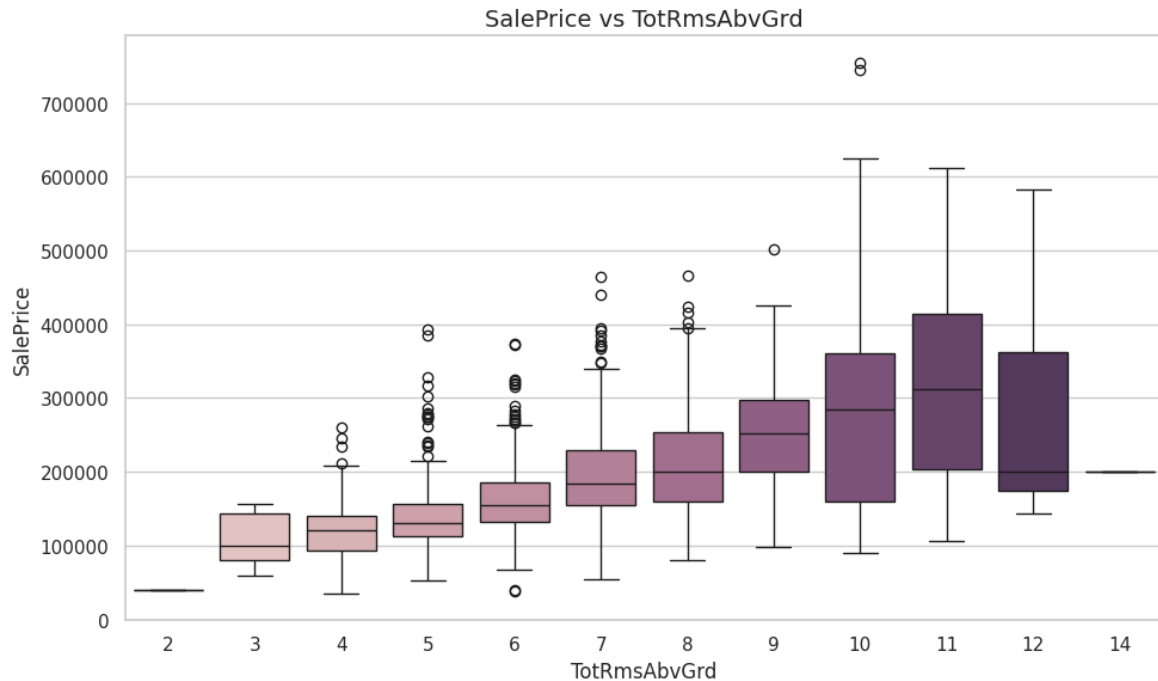
Interpretasi:

Boxplot ini memperlihatkan hubungan antara jumlah kamar mandi penuh (FullBath) dengan harga jual rumah (SalePrice). Secara umum, terlihat bahwa semakin banyak jumlah kamar mandi penuh, semakin tinggi pula harga jual rumah. Rumah dengan 0 kamar mandi penuh memiliki median harga yang paling rendah, sekitar 130.000, dan rentang harga yang sempit. Kategori 1 FullBath memiliki median harga sedikit lebih tinggi, namun menunjukkan banyak outlier yang menandakan adanya variasi harga akibat faktor lain.

Peningkatan signifikan terlihat pada rumah dengan 2 FullBath, di mana median harga mendekati 200.000, serta sebaran harga yang semakin luas. Kategori 3 FullBath memiliki median harga tertinggi, di atas 300.000, dan distribusi harga yang sangat lebar, bahkan mencapai hampir 750.000. Hal ini menunjukkan bahwa rumah dengan 3 kamar mandi penuh umumnya termasuk dalam kategori rumah mewah.

Secara keseluruhan, visualisasi ini menunjukkan adanya korelasi positif antara jumlah kamar mandi penuh dan harga jual rumah. Namun, banyaknya outlier di setiap kategori juga mengindikasikan bahwa harga rumah sangat dipengaruhi oleh kombinasi berbagai faktor lain, seperti ukuran rumah, lokasi, dan kualitas konstruksi.

'TotRmsAbvGrd'



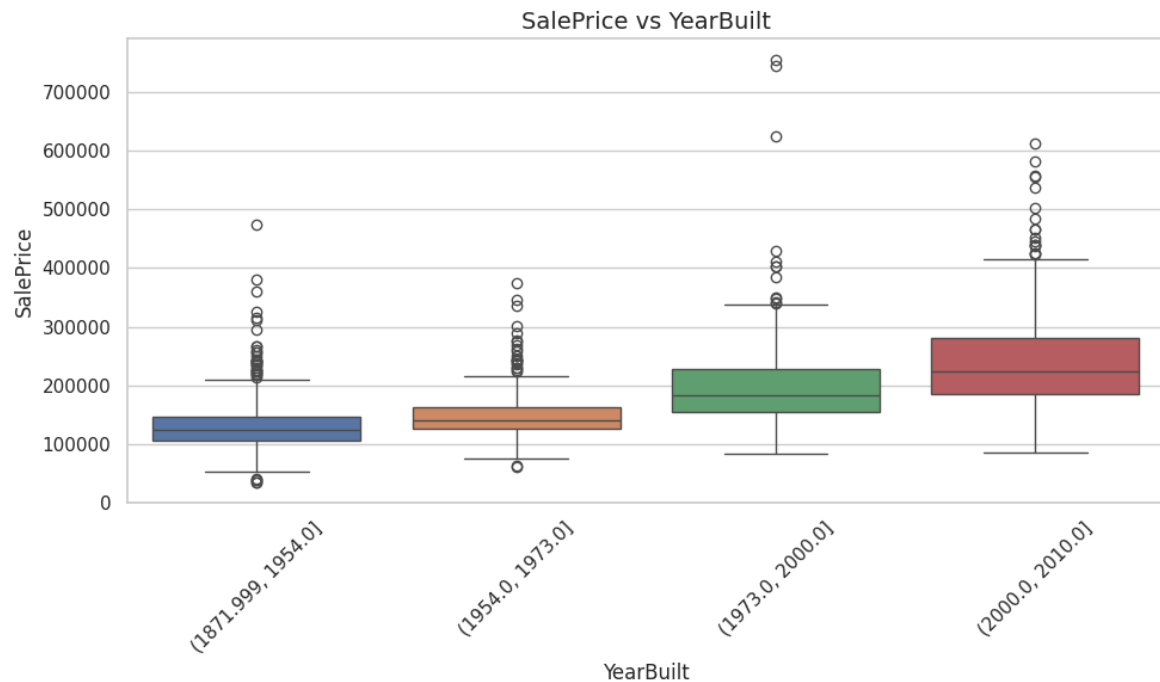
Interpretasi:

Boxplot ini menunjukkan hubungan antara jumlah ruangan di atas tanah (TotRmsAbvGrd) dan harga jual rumah (SalePrice). Terlihat bahwa semakin banyak jumlah ruangan, cenderung semakin tinggi pula harga jual rumah. Pada rumah dengan 2 hingga 4 ruangan, median harga masih rendah, berada di bawah 150.000. Kategori 5 hingga 7 ruangan mulai menunjukkan kenaikan median harga ke kisaran 170.000–220.000, dengan rentang harga yang lebih luas dan banyak outlier.

Kenaikan lebih signifikan terlihat pada rumah dengan 8 hingga 11 ruangan, di mana median harga mendekati atau bahkan melebihi 300.000. Kategori ini juga menunjukkan banyak outlier dengan harga sangat tinggi, mendekati 700.000, yang mencerminkan kemungkinan adanya fitur tambahan atau lokasi premium. Namun, pada jumlah ruangan ekstrem seperti 14, median harga justru turun, yang mungkin disebabkan oleh jumlah data yang sangat sedikit atau rumah tersebut tidak mewakili tren umum.

Secara keseluruhan, terdapat korelasi positif antara jumlah ruangan dan harga rumah, meskipun tidak sepenuhnya linier. Banyaknya outlier juga mengindikasikan bahwa faktor lain tetap berperan besar dalam menentukan harga rumah.

'YearBuilt'



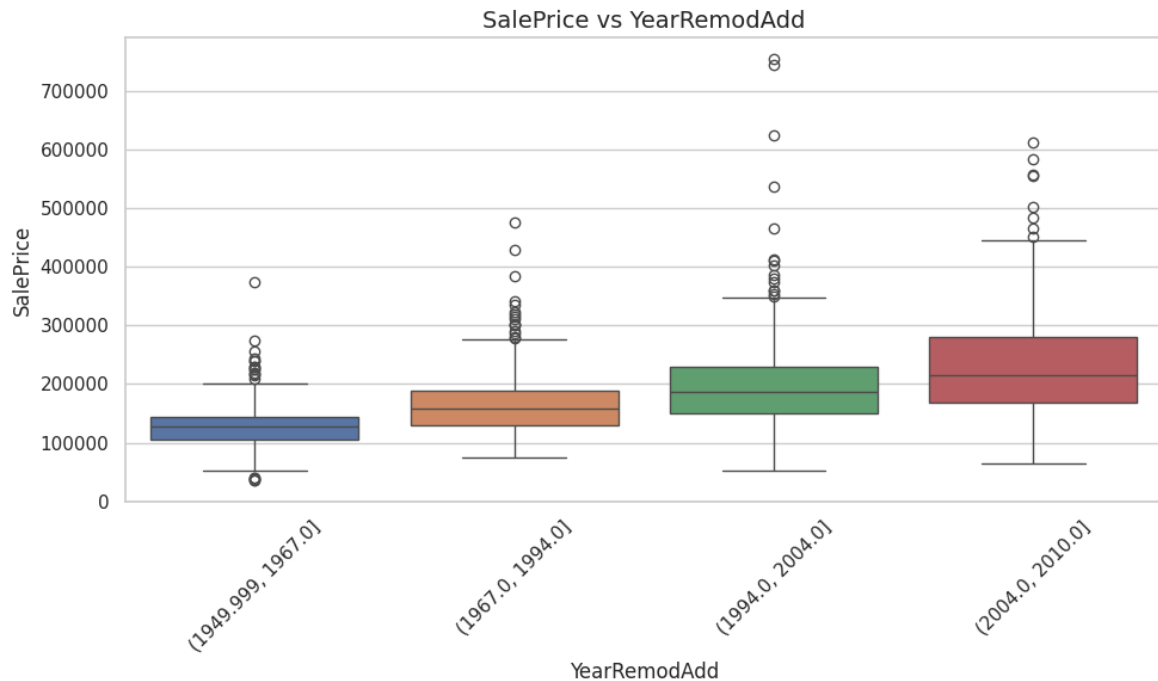
Interpretasi:

Boxplot ini menunjukkan hubungan antara tahun pembangunan rumah (YearBuilt) yang telah dibagi ke dalam empat kelompok periode, dengan harga jual rumah (SalePrice). Terlihat bahwa semakin baru tahun pembangunan, harga jual rumah cenderung semakin tinggi.

Rumah yang dibangun pada periode 1871–1954 dan 1954–1973 memiliki median harga yang paling rendah, berkisar antara 120.000 hingga 140.000, dengan sebaran harga yang relatif sempit. Sementara itu, rumah yang dibangun antara 1973–2000 menunjukkan kenaikan median harga hingga sekitar 180.000, dengan rentang harga yang lebih luas dan mulai muncul banyak outlier. Kategori dengan tahun pembangunan terbaru, yaitu 2000–2010, memiliki median harga tertinggi (sekitar 220.000–240.000) dan distribusi harga yang lebih lebar, serta lebih banyak outlier dengan harga di atas 400.000.

Dari visualisasi ini dapat diinterpretasikan bahwa tahun pembangunan rumah merupakan indikator penting dalam menentukan harga jual. Rumah yang lebih baru cenderung memiliki harga jual yang lebih tinggi, kemungkinan karena desain modern, bahan bangunan yang lebih baik, atau kondisi bangunan yang masih relatif baru. Namun, keberadaan outlier di setiap kategori tetap menunjukkan bahwa faktor lain seperti lokasi, ukuran, dan fasilitas juga turut memengaruhi nilai rumah.

'YearRemodAdd'



Interpretasi:

Boxplot antara YearRemodAdd dan SalePrice menunjukkan bahwa semakin baru tahun renovasi, semakin tinggi harga jual rumah. Rumah yang tidak pernah direnovasi atau direnovasi sejak lama umumnya memiliki harga jual lebih rendah, dengan median di bawah 150.000 dan sebaran harga yang sempit. Sebaliknya, rumah yang direnovasi setelah tahun 2000 cenderung memiliki median harga lebih tinggi, bahkan mendekati atau melebihi 250.000, serta menunjukkan rentang harga yang lebih luas dan banyak outlier. Hal ini mengindikasikan bahwa renovasi rumah berpengaruh signifikan terhadap peningkatan nilai jualnya, meskipun faktor lain seperti lokasi, ukuran, dan kualitas bangunan juga turut memengaruhi.

2.3 Validation 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

BAB 3 DATA PREPARATION

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.

3.1 Data Selection

Code:

```
# Pisahkan fitur dan target
X = train.drop(['Id', 'SalePrice'], axis=1) # Hilangkan ID karena
tidak berguna
y = train['SalePrice']

# Gabungkan data train dan test untuk preprocessing bersama
all_data = pd.concat([X, test.drop(['Id'], axis=1)],
axis=0).reset_index(drop=True)

# Cek ukuran
print("Ukuran gabungan data sebelum preprocessing:",
all_data.shape)
```

Output:

```
Ukuran gabungan data sebelum preprocessing: (2919, 79)
```

Interpretasi:

Mempersiapkan dataset dengan memisahkan antara fitur (X) dan target (y), serta mengidentifikasi tipe data dari setiap fitur (numerik atau kategorikal). Hal ini memungkinkan untuk melakukan pra-pemrosesan data yang sesuai, seperti normalisasi untuk fitur numerik dan encoding untuk fitur kategorikal, sebelum diterapkan pada model machine learning.

3.2 Data Cleaning

Code:

```
# Cek nilai yang hilang
missing = all_data.isnull().sum()
missing = missing[missing > 0].sort_values(ascending=False)

print("\nFitur dengan nilai kosong:")
display(missing)
```

Output:

Fitur dengan nilai kosong:

	0
PoolQC	2909

MiscFeature	2814
Alley	2721
Fence	2348
MasVnrType	1766
FireplaceQu	1420
LotFrontage	486
GarageQual	159
GarageYrBlt	159
GarageCond	159
GarageFinish	159
GarageType	157
BsmtExposure	82
BsmtCond	82
BsmtQual	81

BsmtFinType2	80
BsmtFinType1	79
MasVnrArea	23
MSZoning	4
BsmtFullBath	2
Functional	2
BsmtHalfBath	2
Utilities	2
BsmtFinSF1	1
Exterior2nd	1
Exterior1st	1
Electrical	1
TotalBsmtSF	1
BsmtUnfSF	1

BsmtFinSF2	1
KitchenQual	1
GarageArea	1
GarageCars	1
SaleType	1

dtype: int64

Pada kode di atas, langkah pertama adalah memeriksa keberadaan nilai kosong (missing values) dalam dataset yang digunakan. Fungsi `isnull()` digunakan untuk memeriksa setiap elemen dalam dataset dan menghasilkan nilai boolean (True jika nilai kosong, dan False jika ada data). Kemudian, dengan menggunakan metode `sum()`, dihitung jumlah nilai kosong yang ada pada setiap fitur. Hasilnya disaring dengan kondisi `missing > 0`, yang hanya memilih fitur yang memiliki nilai kosong lebih dari 0, dan kemudian hasilnya disortir dalam urutan menurun berdasarkan jumlah nilai kosong dengan menggunakan `sort values(ascending=False)`. Tujuan dari langkah ini adalah untuk mengidentifikasi fitur mana saja yang memiliki nilai kosong, sehingga kita bisa fokus pada fitur yang membutuhkan penanganan lebih lanjut. Tahap ini hanya sebatas mendeteksi dan tidak melakukan penanganan langsung terhadap nilai kosong, yang mana akan dilakukan pada tahap berikutnya (rekonstruksi atau imputasi). Dengan demikian, hasil yang ditampilkan memberikan gambaran awal mengenai distribusi nilai kosong dalam dataset yang perlu ditangani sebelum data dapat digunakan untuk pelatihan model machine learning.

3.3 Data Construct

```
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler

# Pisahkan fitur numerik dan kategorikal
numeric_features = all_data.select_dtypes(include=['int64',
'float64']).columns.tolist()
categorical_features =
all_data.select_dtypes(include=['object']).columns.tolist()

# Imputasi nilai kosong
# Numerik: Median
num_imputer = SimpleImputer(strategy='median')
```

```
all_data[numeric_features] =
num_imputer.fit_transform(all_data[numeric_features])

# Kategorikal: Modus
cat_imputer = SimpleImputer(strategy='most_frequent')
all_data[categorical_features] =
cat_imputer.fit_transform(all_data[categorical_features])
```

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.

Pada kode di atas, tahap Data Construction menangani nilai kosong (missing values) dengan cara yang berbeda untuk fitur numerik dan kategorikal. Fitur numerik diisi dengan median agar lebih robust terhadap outlier, sementara fitur kategorikal diisi dengan modus (nilai yang paling sering muncul). Selanjutnya, fitur kategorikal diubah menjadi format yang dapat diproses oleh model menggunakan one-hot encoding, dan fitur numerik diskalakan dengan StandardScaler untuk memastikan distribusi data memiliki mean 0 dan standar deviasi 1.

3.4 Labeling Data

```
from sklearn.preprocessing import OneHotEncoder

# One-hot encoding untuk data kategorikal
all_data = pd.get_dummies(all_data, columns=categorical_features)

print("Ukuran data setelah one-hot encoding:", all_data.shape)
```


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

```
# Skalikan data numerik
scaler = StandardScaler()
all_data[numeric_features] =
scaler.fit_transform(all_data[numeric_features])

# Pisahkan kembali data train dan test setelah preprocessing
X_processed = all_data.iloc[:len(X), :]
test_processed = all_data.iloc[len(X):, :]

print("\nUkuran data setelah preprocessing:")
print("X_processed:", X_processed.shape)
print("test_processed:", test_processed.shape)
```

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