```
# Import library yang dibutuhkan
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Atur agar visualisasi tampil di notebook
%matplotlib inline

# Muat dataset
# Ganti 'path/to/your/train.csv' dengan lokasi file Anda
df = pd.read_csv('train.csv')

# Tampilkan 5 baris pertama data
print("5 Baris Pertama Data:")
print(df.head())

# Tampilkan informasi dasar tentang dataset
print("\nInformasi Dataset:")
df.info()
```

```
5 Baris Pertama Data:
   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
0   1          60       RL         65.0     8450   Pave   NaN      Reg
1   2          20       RL         80.0     9600   Pave   NaN      Reg
2   3          60       RL         68.0    11250   Pave   NaN      IR1
3   4          70       RL         60.0     9550   Pave   NaN      IR1
4   5          60       RL         84.0    14260   Pave   NaN      IR1

  LandContour Utilities  ... PoolArea PoolQC Fence MiscFeature MiscVal MoSold  \
0         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      2
1         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      5
2         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      9
3         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      2
4         Lvl    AllPub  ...        0    NaN   NaN         NaN       0     12

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500
3   2006        WD        Abnorml     140000
4   2008        WD         Normal     250000

[5 rows x 81 columns]

Informasi Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
 12  Neighborhood   1460 non-null   object
 13  Condition1     1460 non-null   object
 14  Condition2     1460 non-null   object
 15  BldgType       1460 non-null   object
 16  HouseStyle     1460 non-null   object
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrType     588 non-null    object
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object
```

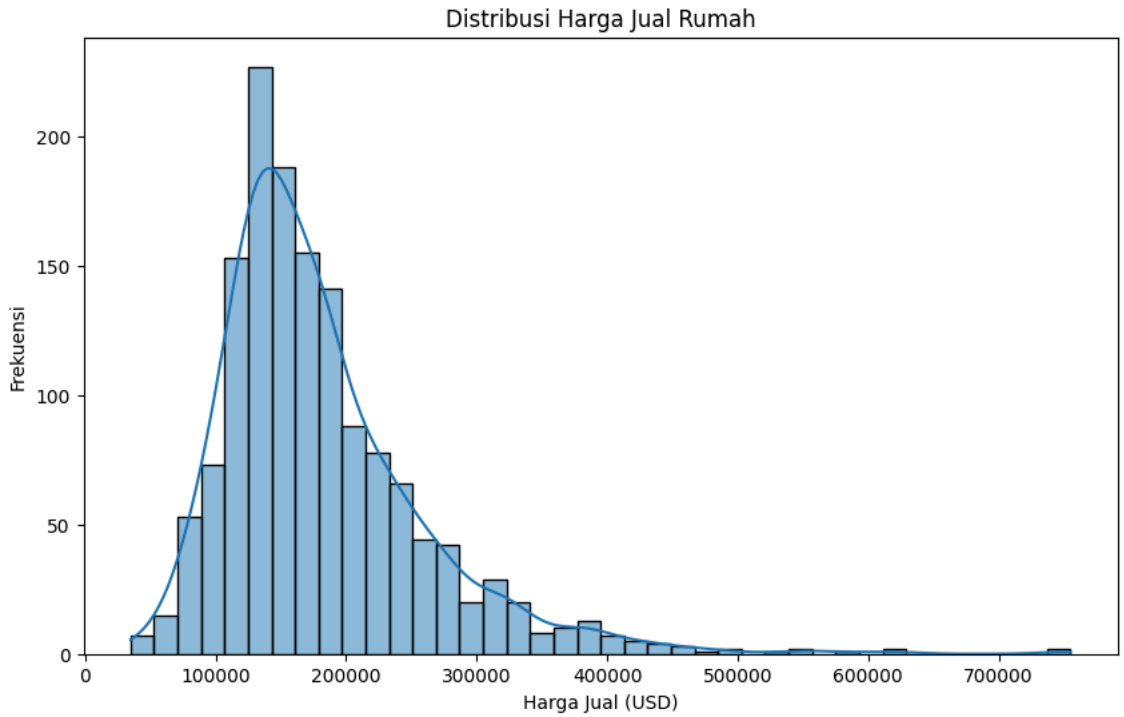
```
# Hitung jumlah missing values di setiap kolom
missing_values = df.isnull().sum().sort_values(ascending=False)
print("Kolom dengan Missing Values:")
print(missing_values[missing_values > 0])
```

```
Kolom dengan Missing Values:
PoolQC          1453
MiscFeature     1406
Alley           1369
Fence           1179
MasVnrType       872
FireplaceQu      690
LotFrontage      259
GarageQual        81
GarageFinish      81
GarageType        81
GarageYrBlt       81
GarageCond        81
BsmtFinType2      38
BsmtExposure      38
BsmtCond          37
BsmtQual          37
BsmtFinType1      37
MasVnrArea         8
Electrical         1
dtype: int64
```

```
# Contoh mengisi missing values
# Kolom 'LotFrontage' (numerik) diisi dengan median
df['LotFrontage'] = df['LotFrontage'].fillna(df['LotFrontage'].median())

# Kolom 'GarageType' (kategorikal) diisi dengan modus
df['GarageType'] = df['GarageType'].fillna(df['GarageType'].mode()[0])

# Untuk semplicitas, kita hapus kolom dengan > 50% data hilang
df.drop(['PoolQC', 'MiscFeature', 'Alley', 'Fence'], axis=1, inplace=True)

# Verifikasi kembali setelah dibersihkan (untuk kolom yang sudah ditangani)
print("\nMissing values setelah penanganan sederhana:")
print(df[['LotFrontage', 'GarageType']].isnull().sum())
```

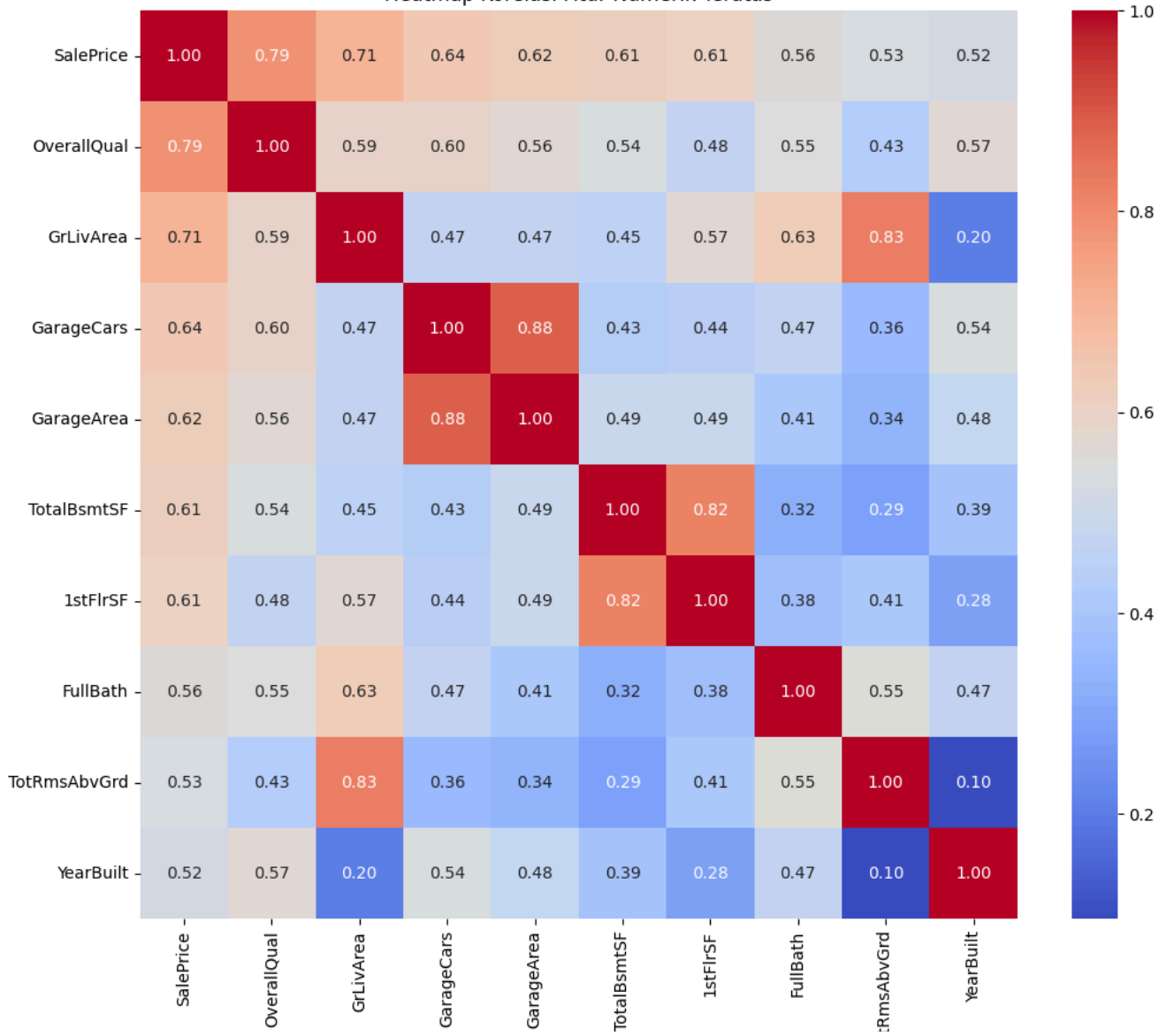
```
Missing values setelah penanganan sederhana:
LotFrontage    0
GarageType     0
dtype: int64
```

```
# Visualisasi distribusi harga rumah
plt.figure(figsize=(10, 6))
sns.histplot(df['SalePrice'], kde=True, bins=40)
plt.title('Distribusi Harga Jual Rumah')
plt.xlabel('Harga Jual (USD)')
plt.ylabel('Frekuensi')
plt.show()

# Tampilkan ringkasan statistik untuk SalePrice
print(df['SalePrice'].describe())
```

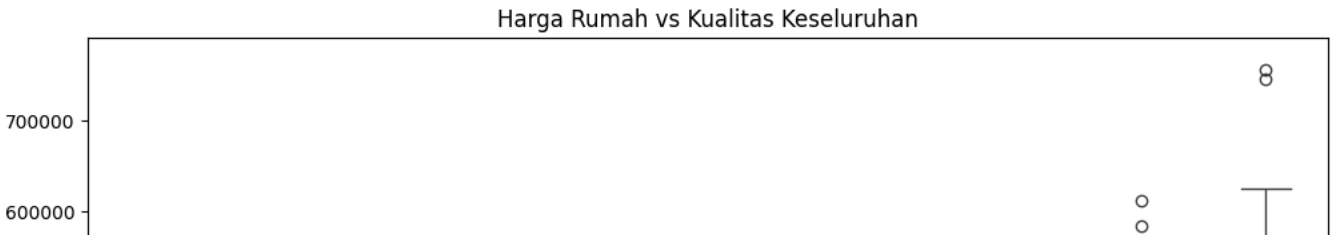



```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

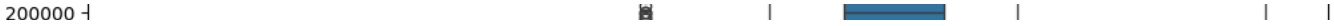
```
# Membuat heatmap korelasi untuk melihat hubungan antar variabel numerik
plt.figure(figsize=(12, 10))
# Select only numeric columns before calculating correlation
numeric_df = df.select_dtypes(include=np.number)
# Pilih 10 fitur dengan korelasi tertinggi dengan SalePrice
korelasi_tertinggi = numeric_df.corr().nlargest(10, 'SalePrice')['SalePrice'].index
matriks_korelasi = numeric_df[korelasi_tertinggi].corr()
sns.heatmap(matriks_korelasi, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Heatmap Korelasi Fitur Numerik Teratas')
plt.show()
```




```
# Boxplot untuk melihat hubungan antara kualitas keseluruhan ('OverallQual') dan harga
plt.figure(figsize=(12, 7))
sns.boxplot(x='OverallQual', y='SalePrice', data=df)
plt.title('Harga Rumah vs Kualitas Keseluruhan')
plt.xlabel('Kualitas Keseluruhan (1-10)')
plt.ylabel('Harga Jual (USD)')
plt.show()
```




```
features = ['OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea', 'TotalBsmtSF', 'FullBath', 'YearBuilt']
target = 'SalePrice'

X = df[features]
y = df[target]

# Contoh jika kita ingin memasukkan 'GarageType'
# X = pd.get_dummies(X, columns=['GarageType'], drop_first=True)

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```




```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

# 1. Model Regresi Linear
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# 2. Model Random Forest
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```