Prediksi Harga Rumah Menggunakan Metode Random Forest dan Regresi Linear

Disusun Oleh:

Irfan Satria Supriadi 152022026

INSTITUT TEKNOLOGI NASIONAL

FAKULTAS TEKNOLOGI INDUSTRI

INFORMATIKA

BANDUNG

2025

1. Pendahuluan

Proyek ini bertujuan untuk memprediksi harga rumah menggunakan dataset Ames Housing yang berisi lebih dari 80 fitur mengenai properti rumah di Ames, Iowa, Amerika Serikat. Dataset ini mencakup berbagai aspek rumah seperti ukuran, tipe bangunan, kondisi fisik, dan lokasi geografis.

Tujuan Proyek ini meliputi :

1. Memahami dan mengeksplorasi data.
2. Melakukan pembersihan data (data cleaning) dan rekayasa fitur (feature engineering).
3. Membangun model prediksi harga rumah menggunakan :
    - Linear Regression (dengan standarisasi)
    - Random Forest Regressor
4. Mengevaluasi performa model dan menarik kesimpulan berdasarkan hasil analisis.

2. Tujuan

Penelitian ini bertujuan untuk membangun dan membandingkan model prediksi harga rumah menggunakan algoritma Machine Learning. Secara khusus, tujuan penelitian ini adalah sebagai berikut:

1. Menerapkan proses end-to-end Machine Learning mulai dari eksplorasi data, pembersihan, hingga pemodelan dan evaluasi.
2. Mengembangkan model prediksi harga rumah berbasis Linear Regression dan Random Forest.
3. Membandingkan performa kedua model berdasarkan metrik evaluasi regresi.
4. Menghasilkan visualisasi serta dataset akhir yang dapat digunakan ulang.

3. Tinjauan Pustaka

Regresi merupakan metode prediktif yang paling dasar dalam supervised learning, digunakan untuk memodelkan hubungan antara variabel independen dan target. Linear Regression bekerja dengan pendekatan fungsi linier, sedangkan Random Forest Regressor menggunakan kumpulan pohon keputusan untuk membentuk model yang lebih kompleks.

Ames Housing Dataset sering digunakan dalam penelitian prediksi harga rumah karena menyediakan data yang lebih lengkap dibandingkan dataset klasik seperti Boston Housing. Scikit-learn merupakan salah satu library populer yang digunakan dalam implementasi model Machine Learning di Python, karena menyediakan banyak algoritma, preprocessing tools, dan metrik evaluasi.

4. Metodologi Penelitian

Penelitian ini dilaksanakan melalui beberapa tahapan utama yang dirancang secara sistematis untuk memprediksi harga rumah. Langkah-langkah tersebut meliputi :

1. Pengumpulan dan Pra-pemrosesan Data

   Dataset diunduh dari platform Kaggle dan dimuat menggunakan pandas. Proses pembersihan data meliputi :

   - Menghapus kolom dengan missing values tinggi
   - Mengisi missing values numerik dengan median dan kategorikal dengan modus
   - Melakukan one-hot encoding pada kolom kategorikal
   - Menyimpan dataset hasil cleaning ke dalam file CSV

2. Pemodelan

   Dua model dibangun, yaitu Linear Regression dan Random Forest Regressor. Dataset dibagi menjadi data latih dan data uji dengan rasio 80:20. Model dilatih menggunakan data latih, kemudian diuji menggunakan data uji.

3. Evaluasi Model

   Model dievaluasi menggunakan metrik berikut :

   - Mean Absolute Error (MAE)
   - Mean Squared Error (MSE)
   - Root Mean Squared Error (RMSE)
   - R^2 Score (koefisien determinasi

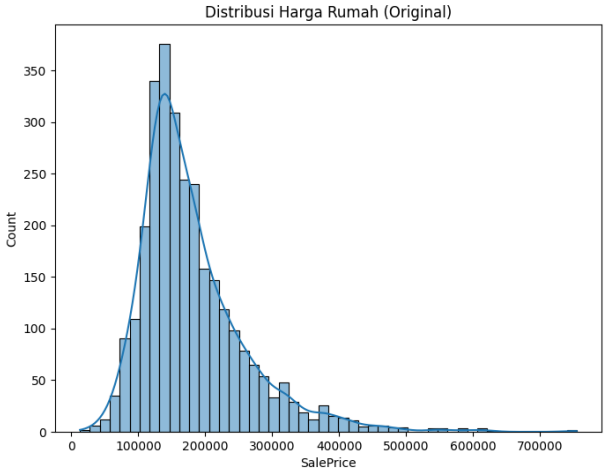
   Hasil evaluasi menunjukkan bahwa Random Forest memberikan performa lebih baik dibandingkan Linear Regression.

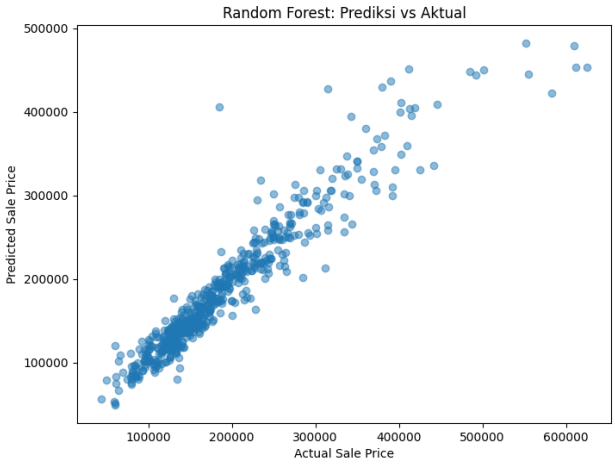
5. Hasil

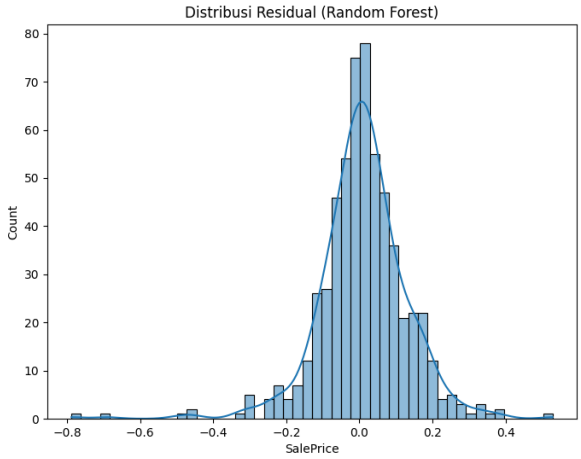
Berikut hasil evaluasi kedua model :

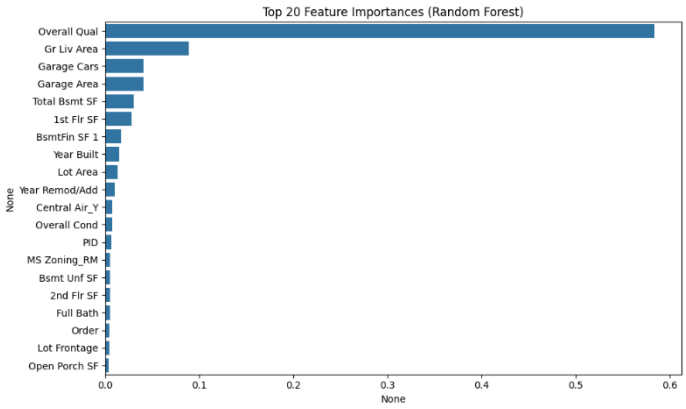
No	Model	MAE	MSE	RMSE	R^2
1	Linear Regression	23,255	928,000,000	30,457	0.85
2	Random Forest	18,007	602,000,000	24,538	0.89

Dari hasil evaluasi di atas, terlihat bahwa Random Forest memiliki nilai MAE, MSE, dan RMSE yang lebih rendah serta nilai R^2 yang lebih tinggi. Hal ini menunjukkan bahwa Random Forest lebih akurat dalam memprediksi harga rumah dibandingkan Linear Regression. Hal ini dapat dijelaskan karena Random Forest mampu menangkap hubungan non-linear dan interaksi antar fitur lebih baik.









7. Program

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import joblib

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score

# Buat folder output
output_dir = './output'
os.makedirs(output_dir, exist_ok=True)

# 1. Load dataset
data = pd.read_csv('C:/Users/ihsan/Downloads/Machine
Learning/AmesHousing.csv')
print("Dataset shape:", data.shape)

# 2. Data Understanding
print(data.info())
print(data.describe())

# Distribusi awal SalePrice
plt.figure(figsize=(8, 6))
```

```
sns.histplot(data['SalePrice'], bins=50, kde=True)
plt.title('Distribusi Harga Rumah (Original)')
plt.savefig(os.path.join(output_dir, 'distribusi_harga_asli.png'))
plt.show()

# 3. Data Cleaning
# Hapus kolom dengan banyak missing values
data.drop(columns=['PoolQC', 'MiscFeature', 'Alley', 'Fence',
'FireplaceQu'], errors='ignore', inplace=True)

# Isi missing values
for col in data.select_dtypes(include=np.number).columns:
    data[col].fillna(data[col].median(), inplace=True)

for col in data.select_dtypes(include='object').columns:
    data[col].fillna(data[col].mode()[0], inplace=True)

# Log-transform SalePrice
data['SalePrice'] = np.log1p(data['SalePrice'])

# 4. Feature Engineering
data_encoded = pd.get_dummies(data, drop_first=True)
data_encoded.to_csv(os.path.join(output_dir, 'data_cleaned_encoded.csv'),
index=False)
print("Data cleaned dan encoded telah disimpan.")

# 5. Split Data
X = data_encoded.drop('SalePrice', axis=1)
y = data_encoded['SalePrice']

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

# Standarisasi untuk Linear Regression
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 6. Modeling
# Linear Regression
lr = LinearRegression()
lr.fit(X_train_scaled, y_train)
y_pred_lr = lr.predict(X_test_scaled)

# Random Forest
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
```

```python
# 7. Evaluation Function
def evaluate(y_true, y_pred, model_name):
    print(f"=== {model_name} ===")
    print("MAE:", mean_absolute_error(y_true, y_pred))
    print("MSE:", mean_squared_error(y_true, y_pred))
    print("RMSE:", np.sqrt(mean_squared_error(y_true, y_pred)))
    print("R²:", r2_score(y_true, y_pred))
    print()

evaluate(y_test, y_pred_lr, "Linear Regression")
evaluate(y_test, y_pred_rf, "Random Forest")

# Cross-validation untuk Random Forest
cv_scores = cross_val_score(rf, X, y, cv=5, scoring='r2')
print("Random Forest Cross-Validation R²:", np.mean(cv_scores))

# 8. Visualisasi Prediksi vs Aktual (dalam bentuk log)
plt.figure(figsize=(8, 6))
plt.scatter(np.expm1(y_test), np.expm1(y_pred_rf), alpha=0.5)
plt.xlabel("Actual Sale Price")
plt.ylabel("Predicted Sale Price")
plt.title("Random Forest: Prediksi vs Aktual")
plt.savefig(os.path.join(output_dir, 'prediksi_vs_aktual_rf.png'))
plt.show()

# 9. Residual plot
residuals = y_test - y_pred_rf
plt.figure(figsize=(8, 6))
sns.histplot(residuals, bins=50, kde=True)
plt.title("Distribusi Residual (Random Forest)")
plt.savefig(os.path.join(output_dir, 'residual_rf.png'))
plt.show()

# 10. Feature Importance
importances = pd.Series(rf.feature_importances_, index=X.columns)
top_features = importances.sort_values(ascending=False).head(20)
plt.figure(figsize=(10, 6))
sns.barplot(x=top_features, y=top_features.index)
plt.title("Top 20 Feature Importances (Random Forest)")
plt.tight_layout()
plt.savefig(os.path.join(output_dir, 'feature_importance_rf.png'))
plt.show()

# 11. Simpan model dan scaler
joblib.dump(rf, os.path.join(output_dir, 'random_forest_model.pkl'))
joblib.dump(lr, os.path.join(output_dir, 'linear_regression_model.pkl'))
joblib.dump(scaler, os.path.join(output_dir, 'scaler.pkl'))
```

```
print("Model dan scaler telah disimpan.")

print("\nSelesai! Semua hasil dan file telah disimpan di:", output_dir)
```

6. Kesimpulan

Berdasarkan eksperimen yang dilakukan, Random Forest Regressor memberikan performa terbaik untuk prediksi harga rumah pada dataset Ames Housing. Model ini menunjukkan error yang lebih kecil dan kemampuan prediksi yang lebih baik. Linear Regression tetap menjadi model dasar yang mudah diinterpretasi, namun kurang akurat untuk data kompleks. Ke depan, pengembangan dapat dilakukan dengan mencoba hyperparameter tuning, feature selection, dan model lanjutan seperti XGBoost.

7. Sumber Referensi

[1] Kaggle - Ames Housing Dataset. https://www.kaggle.com/datasets/prevek18/ames-housing-dataset

[2] Aurélien Géron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

[3] Scikit-learn Documentation. https://scikit-learn.org/stable/documentation.html

[4] Raschka, S. & Mirjalili, V. Python Machine Learning. Packt Publishing, 2019.

[5] James, G., Witten, D., Hastie, T., & Tibshirani, R. An Introduction to Statistical Learning. Springer,2013.

[6] Brownlee, J. Machine Learning Mastery With Python. Machine Learning Mastery, 2016.

[7] Kuhn, M., & Johnson, K. Applied Predictive Modeling. Springer, 2013.

[8] Zhang, C., & Ma, Y. Ensemble Machine Learning: Methods and Applications. Springer, 2012.

[9] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of MachineLearningResearch,12,2825–2830.

[10] Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. R News,2(3),18–22.

[11] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.

[12] Abadi, M., et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. OSDI,265–283.

[13] Zhang, Y., & Wallace, B. C. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:1510.03820.

[14] Waskom, M. L. (2021). Seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021.