WEEKLY ASSIGNMENT 1

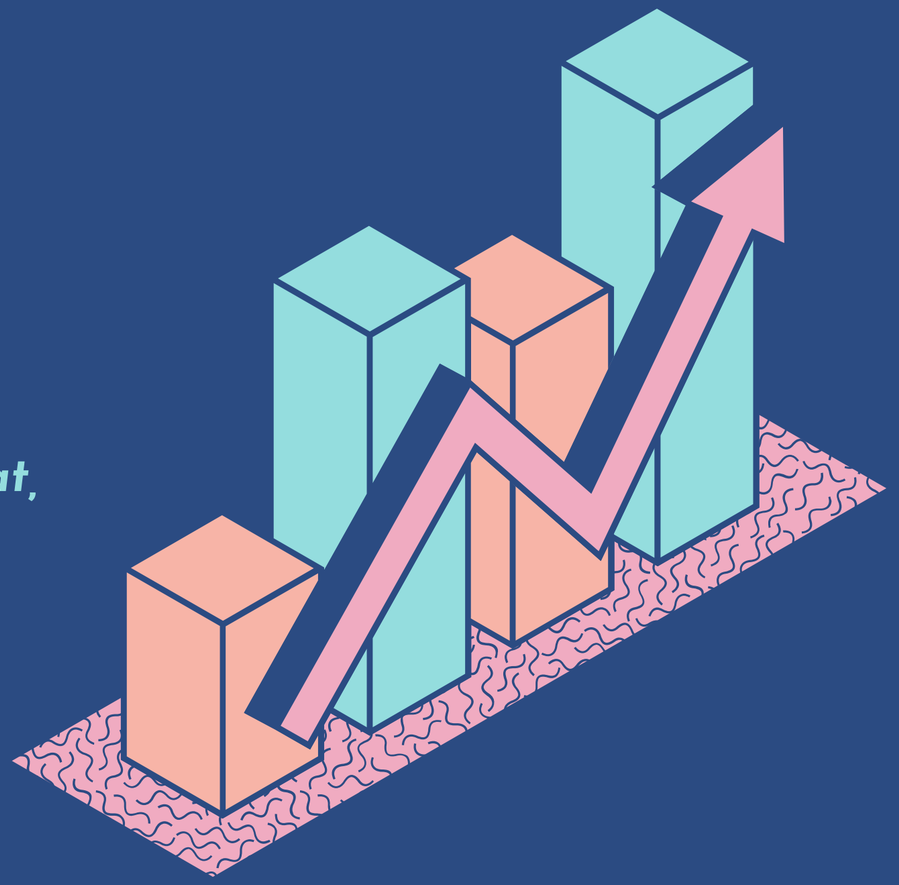
House Pricing Prediction

Industri: Real Estate

Kelompok: CV2 Oppenheimer

Seperti apa industri Real Estate di Indonesia?

Industri Real Estate di Indonesia mengalami pertumbuhan pesat, terutama di perkotaan, dengan proyek-proyek hunian dan komersial yang terus berkembang.



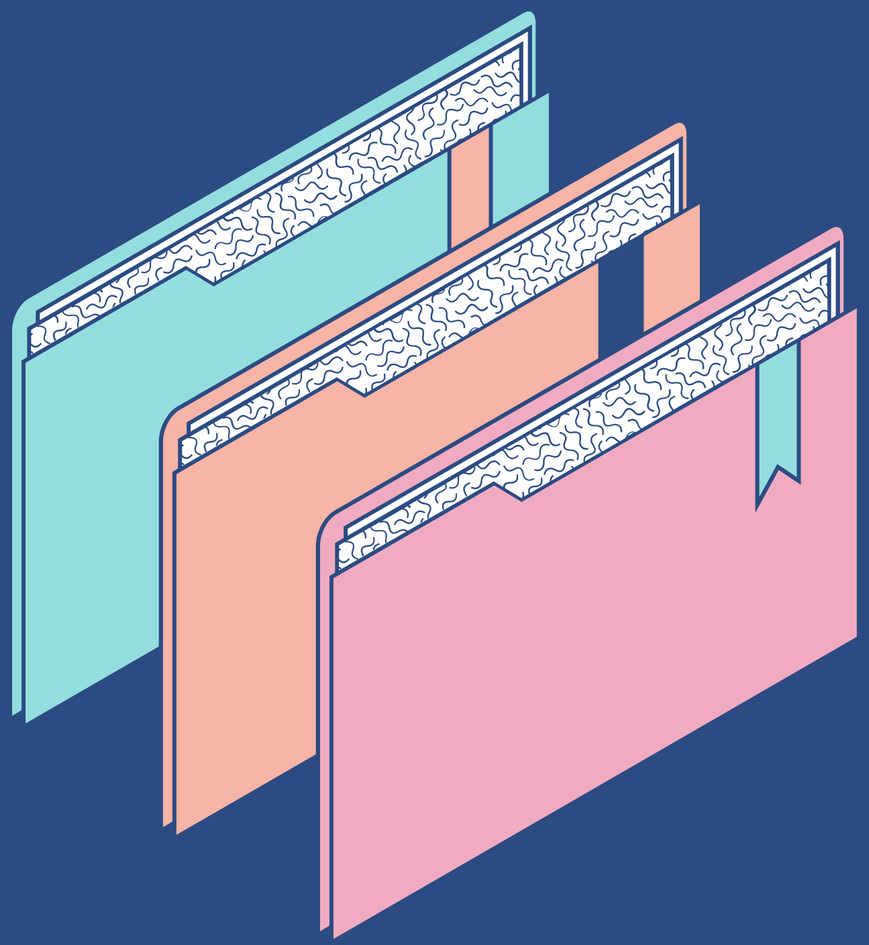
Seberapa penting sebuah sistem yang mampu menentukan harga rumah di industri tersebut?

Sebuah sistem yang mampu menentukan harga rumah sangat penting dalam industri Real Estate karena membantu penjual, pembeli, dan investor membuat keputusan yang lebih informasional dan akurat, menghindari overpricing atau underpricing, dan meningkatkan efisiensi pasar.

Dataset

AMES Housing Data

Data harga rumah di kawasan Ames, Lowa, US
Terdiri dari 1.460 contoh rumah dengan 79 fitur



Preprocessing

- *Handling missing values*
- *Encoding*
- *Scaling*

Missing Values

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

```
PoolQC: 1456
MiscFeature: 1408
Alley: 1352
Fence: 1169
FireplaceQu: 730
LotFrontage: 227
GarageYrBlt: 78
GarageFinish: 78
GarageQual: 78
GarageCond: 78
GarageType: 76
BsmtCond: 45
BsmtExposure: 44
BsmtQual: 44
BsmtFinType2: 42
BsmtFinType1: 42
MasVnrType: 16
MasVnrArea: 15
MSZoning: 4
Functional: 2
BsmtHalfBath: 2
BsmtFullBath: 2
Utilities: 2
KitchenQual: 1
SaleType: 1
BsmtFinSF1: 1
GarageCars: 1
BsmtUnfSF: 1
TotalBsmtSF: 1
Exterior2nd: 1
Exterior1st: 1
GarageArea: 1
BsmtFinSF2: 1
```

Data Description

Asumsi missing values

- **sebagai 1 class kosong (categorical)**
- **bernilai 0 (numerical)**

```
'LotFrontage': 0,
'Alley': 'NA',
'MasVnrType': 'None',
'MasVnrArea': 0,
'BsmtQual': 'NA',
'BsmtCond': 'NA',
'BsmtExposure': 'NA',
'BsmtFinType1': 'NA',
'BsmtFinType2': 'NA',
'FireplaceQu': 'NA',
'GarageType': 'NA',
'GarageFinish': 'NA',
'GarageQual': 'NA',
'GarageCond': 'NA',
'PoolQC': 'NA',
'Fence': 'NA',
'MiscFeature': 'NA',
'Exterior1st': 'Other',
'Exterior2nd': 'Other',
'BsmtFinSF1': 0,
'BsmtFinSF2': 0,
'BsmtUnfSF': 0,
'TotalBsmtSF': 0,
'BsmtFullBath': 0,
'BsmtHalfBath': 0,
'GarageCars': 0,
'GarageArea': 0,
'SaleType': 'Oth'
```

Train Val Split

test size = 25%

```
[ ]  train_df, val_df = train_test_split(df, test_size=0.25, random_state=42)
```

```
train_df
```

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	Lan
1023	120	RL	43.0	3182	Pave	NA	Reg	Lvl	AllPub	Inside	
810	20	RL	78.0	10140	Pave	NA	Reg	Lvl	AllPub	Inside	
1384	50	RL	60.0	9060	Pave	NA	Reg	Lvl	AllPub	Inside	
626	20	RL	0.0	12342	Pave	NA	IR1	Lvl	AllPub	Inside	
813	20	RL	75.0	9750	Pave	NA	Reg	Lvl	AllPub	Inside	
...	...	...	...	...	...	...	...	...	...	...	
1095	20	RL	78.0	9317	Pave	NA	IR1	Lvl	AllPub	Inside	
1130	50	RL	65.0	7804	Pave	NA	Reg	Lvl	AllPub	Inside	
1294	20	RL	60.0	8172	Pave	NA	Reg	Lvl	AllPub	Inside	
860	50	RL	55.0	7642	Pave	NA	Reg	Lvl	AllPub	Corner	
1126	120	RL	53.0	3684	Pave	NA	Reg	Lvl	AllPub	Inside	

1095 rows × 80 columns

Missing Values (Numerical)

```
sns.distplot(train_df['GarageYrBlt'])
plt.show()
```

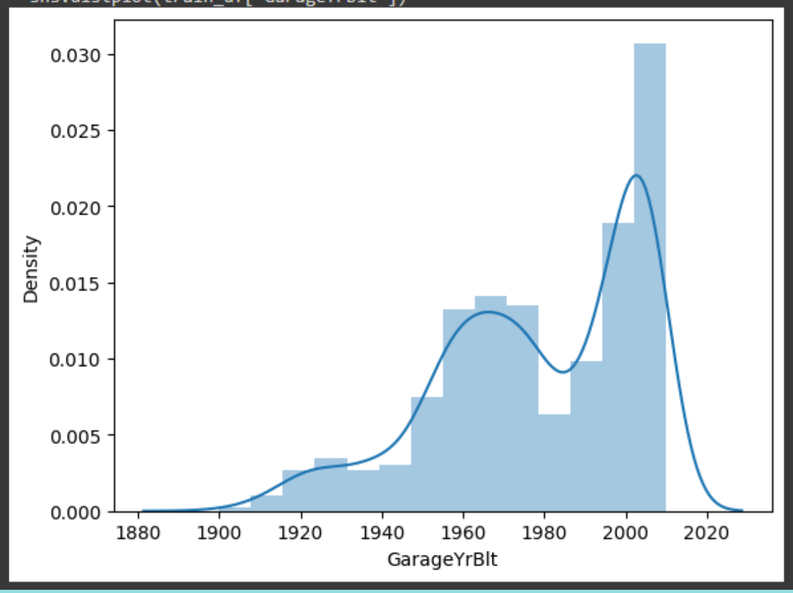
```
<ipython-input-19-36257747dc2a>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(train_df['GarageYrBlt'])
```



distribusi skewed -> impute median

```
impute_num_cols = {'GarageYrBlt': train_df['GarageYrBlt'].median()}
impute_num_cols
```

```
{'GarageYrBlt': 1980.0}
```

Missing Values (Categorical)

'Electrical', 'MSZoning', 'Utilities', 'KitchenQual', 'Functional'

impute mode

```
missing_cat_cols = ['Electrical', 'MSZoning', 'Utilities', 'KitchenQual', 'Functional']
impute_cat_cols = {}
for col in missing_cat_cols:
    impute_cat_cols[col] = train_df[col].mode()[0]
    print(train_df[col].value_counts())
    print('****************************')
```

Feature Selection

Remove Categorical column yang hampir semua terdiri dari 1 class saja

```
******************************
NA    1089
Fa       2
Ex       2
Gd       2
Name: PoolQC, dtype: int64
******************************
```

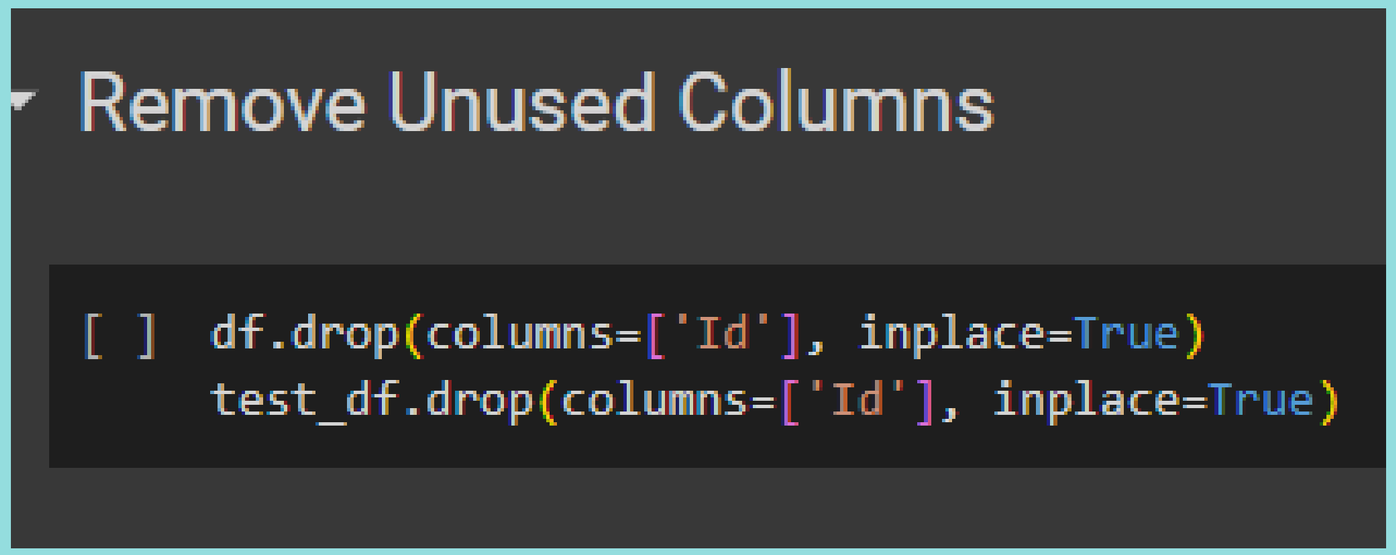
Drop categorical columns that almost all belong to 1 class

```
[38] dropped_cat_cols = ['Street', 'Alley', 'LandContour', 'LandSlope', 'Condition2', 'RoofMatl', 'Heating', 'PoolQC', 'MiscFeature']

     train_df = train_df.drop(columns=dropped_cat_cols)
     val_df = val_df.drop(columns=dropped_cat_cols)
     test_df = test_df.drop(columns=dropped_cat_cols)
```

Feature Selection

Remove Id column karena unique untuk setiap row



Encoding

One Hot Encoding untuk categorical columns

Contoh beberapa kolom hasil encoding

Shape_IR3	LotShape_Reg	Utilities_NoSeWa	LotConfig_CulDSac	LotConfig_FR2	LotConfig_FR3	LotConfig_Inside	Neighborhood_Blueste	N
0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	
0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	
0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	
0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	

1095 baris dan 224 feature

Scaling

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF
0	1.475911	-0.409689	-0.683950	0.637073	-0.515364	1.107889	1.020374	-0.519303	-0.944261	-0.284678	1.711873
1	-0.871228	0.576699	-0.054883	-0.094926	0.390453	0.094543	0.682585	-0.023289	0.469362	2.166141	-1.279161
2	-0.167086	0.069414	-0.152524	-0.094926	-0.515364	-1.049557	-1.681937	-0.601000	-0.533502	-0.284678	-0.478553
3	-0.871228	-1.621537	0.144198	-0.826925	-0.515364	-0.363097	-0.330782	-0.601000	-0.979219	-0.284678	0.920261
4	-0.871228	0.492152	-0.090142	-0.094926	0.390453	-0.428474	-1.295893	0.817019	0.349193	-0.284678	0.596420

Training

Optimize Hyperparamater

```
# Create a RandomForestRegressor instance
rf_regressor = RandomForestRegressor(random_state=42)

# Define the hyperparameter grid for grid search
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

```
# Create the GridSearchCV object with RMSE as the scoring metric
grid_search = GridSearchCV(
    rf_regressor,
    param_grid,
    scoring=rmse_scorer,
    cv=5,  # Cross-validation folds
    verbose=4,  # Enable verbose output for progress
    n_jobs=-1  # Use all available CPU cores for parallel processing
)
```

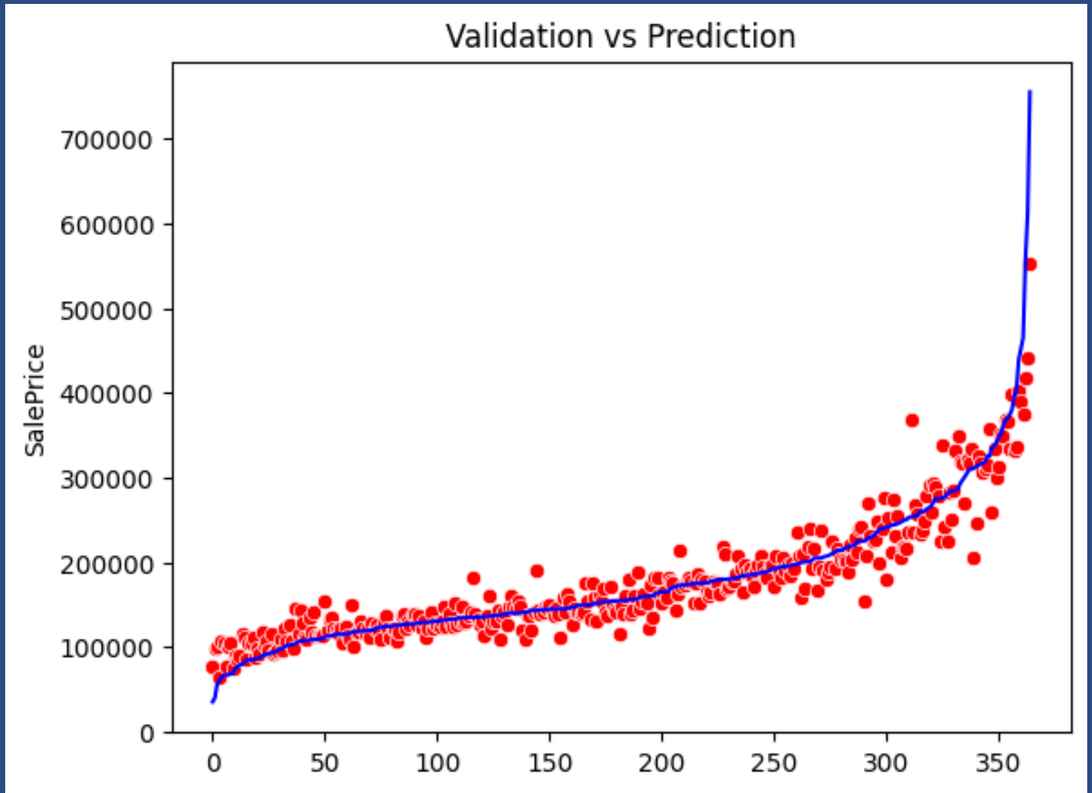
```
Best Hyperparameters: {'max depth': 20, 'min samples leaf': 2, 'min samples split': 10, 'n estimators': 200}
```

Training

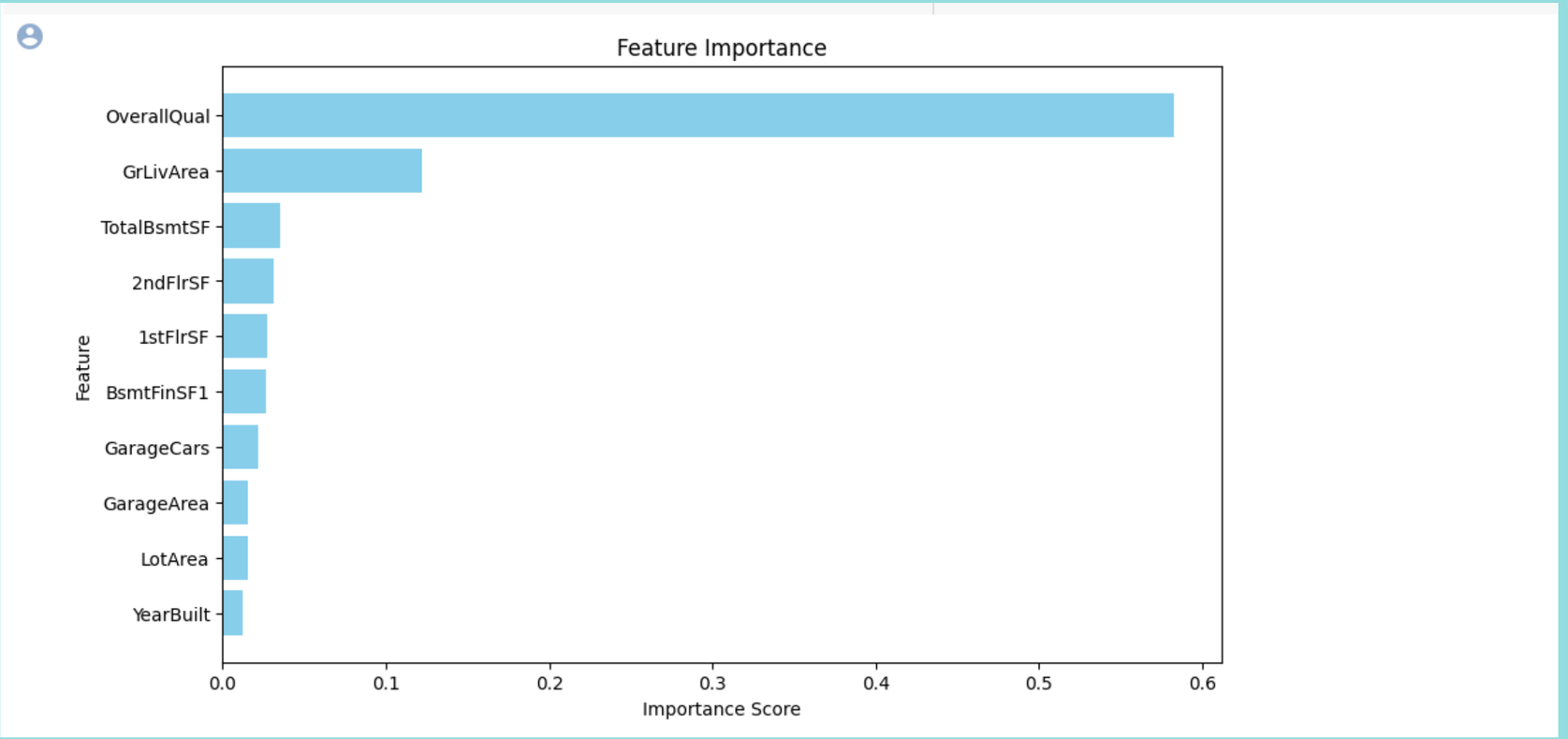
Evaluation

```
y_pred = best_rf_model.predict(X_val)
rmse = np.sqrt(mean_squared_error(msc.inverse_transform(y_val), msc.inverse_transform(y_pred.reshape(-1, 1))))
```

Nilai RMSE : 27658.040281



Feature Importance




```
[ ]  importance_features = importance_df[importance_df['Importance'] > 0.01]['Feature'].tolist()
     importance_features
```

```
['OverallQual',
 'GrLivArea',
 'TotalBsmtSF',
 '2ndFlrSF',
 '1stFlrSF',
 'BsmtFinSF1',
 'GarageCars',
 'GarageArea',
 'LotArea',
 'YearBuilt']
```

Training

Retraining Using Feature Importance

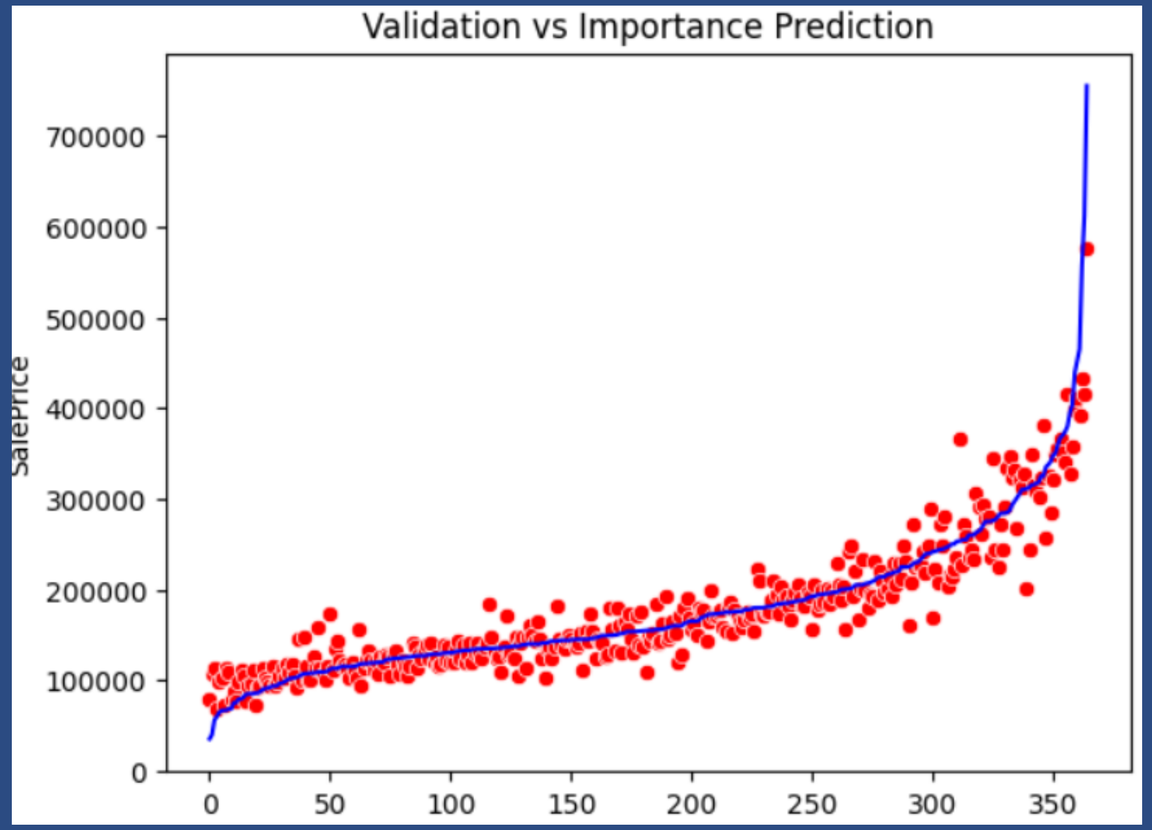
```
X_train_imp = X_train_df[importance_features]
X_val_imp = X_val_df[importance_features]
```

```
grid_search.best_params_
```

```
{'max_depth': 20,
 'min_samples_leaf': 2,
 'min_samples_split': 10,
 'n_estimators': 200}
```

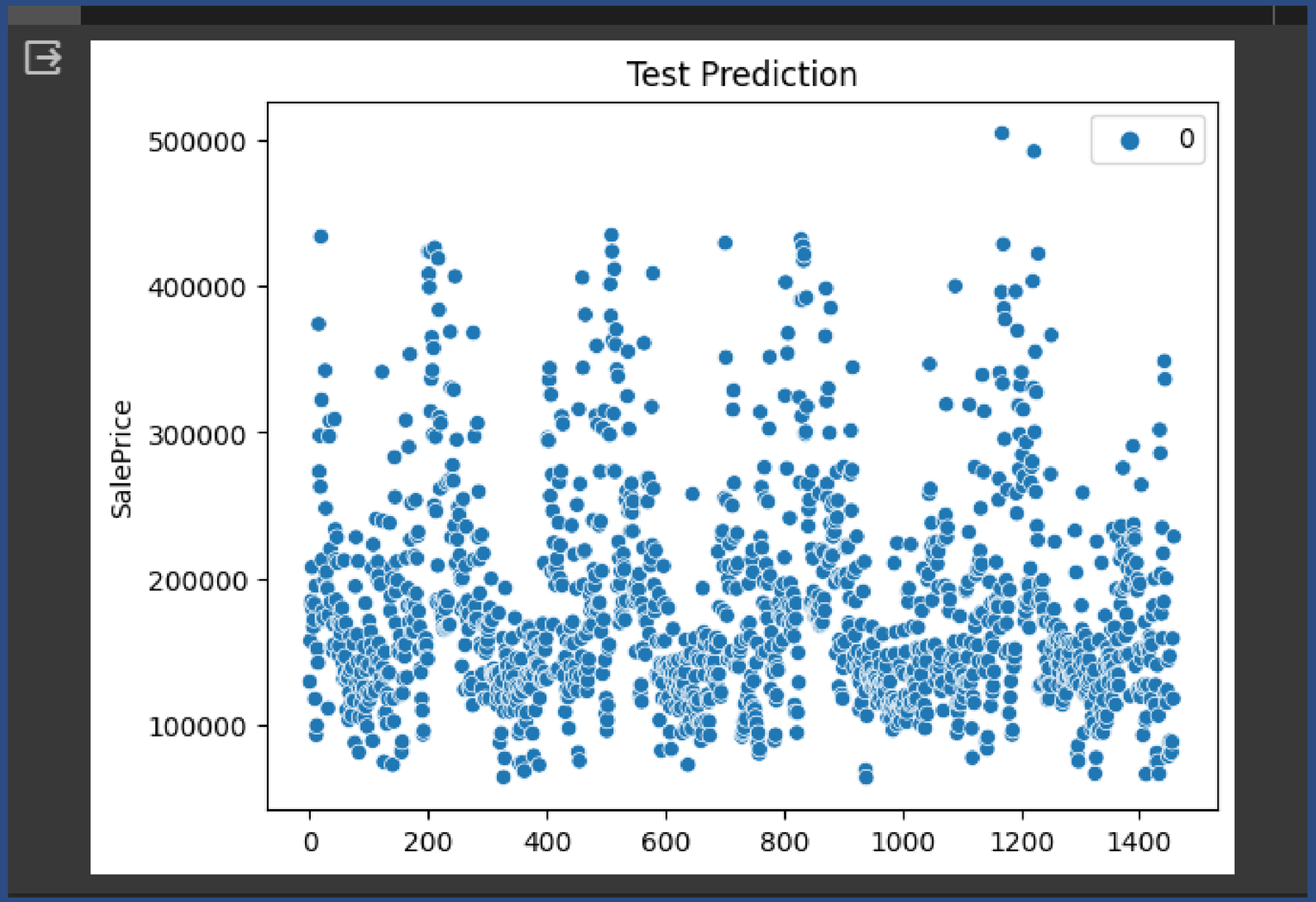
```
rf_imp_model = RandomForestRegressor(n_estimators=200, max_depth=20, min_samples_leaf=2, min_samples_split=10)
```

	Feature	Importance
3	OverallQual	0.583059
15	GrLivArea	0.122051
11	TotalBsmtSF	0.035345
13	2ndFlrSF	0.030950
12	1stFlrSF	0.027423
8	BsmtFinSF1	0.026775
25	GarageCars	0.022129
26	GarageArea	0.015287
2	LotArea	0.015181
5	YearBuilt	0.012318



Nilai RMSE : 28405.57182480416

Test


Test Prediction
SalePrice
500000
400000
300000
200000
100000
0
200
400
600
800
1000
1200
1400
0

Kesimpulan

- *OverallQual:* Rates the overall material and finish of the house menjadi fitur yang paling penting dalam prediksi SalePrice
- *GrLivArea*: Above grade (ground) living area square feet juga menjadi fitur yang penting dalam prediksi SalePrice

Thank you