



KELOMPOK 6



LINEAR DAN POLYNOMIAL REGRESSION

ANGGOTA

- Alhusna Hanifah
- Iffatun Nisa Nasrullah
- Qandila Ahmara
- Muhammad Bintang Indra Hidayat
- Farhanul Khair

PEMAHAMAN DATASET

1. Deskripsi Dataset

Dataset berfokus pada efisiensi energi bangunan, dengan tujuan menganalisis kebutuhan heating load (beban pemanasan) dan cooling load (beban pendinginan) berdasarkan parameter bangunan. Data dikumpulkan melalui simulasi 12 bentuk bangunan berbeda menggunakan perangkat lunak Ecotect, dengan variasi seperti luas kaca, distribusi area kaca, dan orientasi bangunan.

2. Variable dalam Data

- X1 Relative Compactness
- X2 Surface Area
- X3 Wall Area
- X4 Roof Area
- X5 Overall Height
- X6 Orientation
- X7 Glazing Area
- X8 Glazing Area Distribution
- Y1 Heating Load
- Y2 Cooling Load



3. Statistik Data

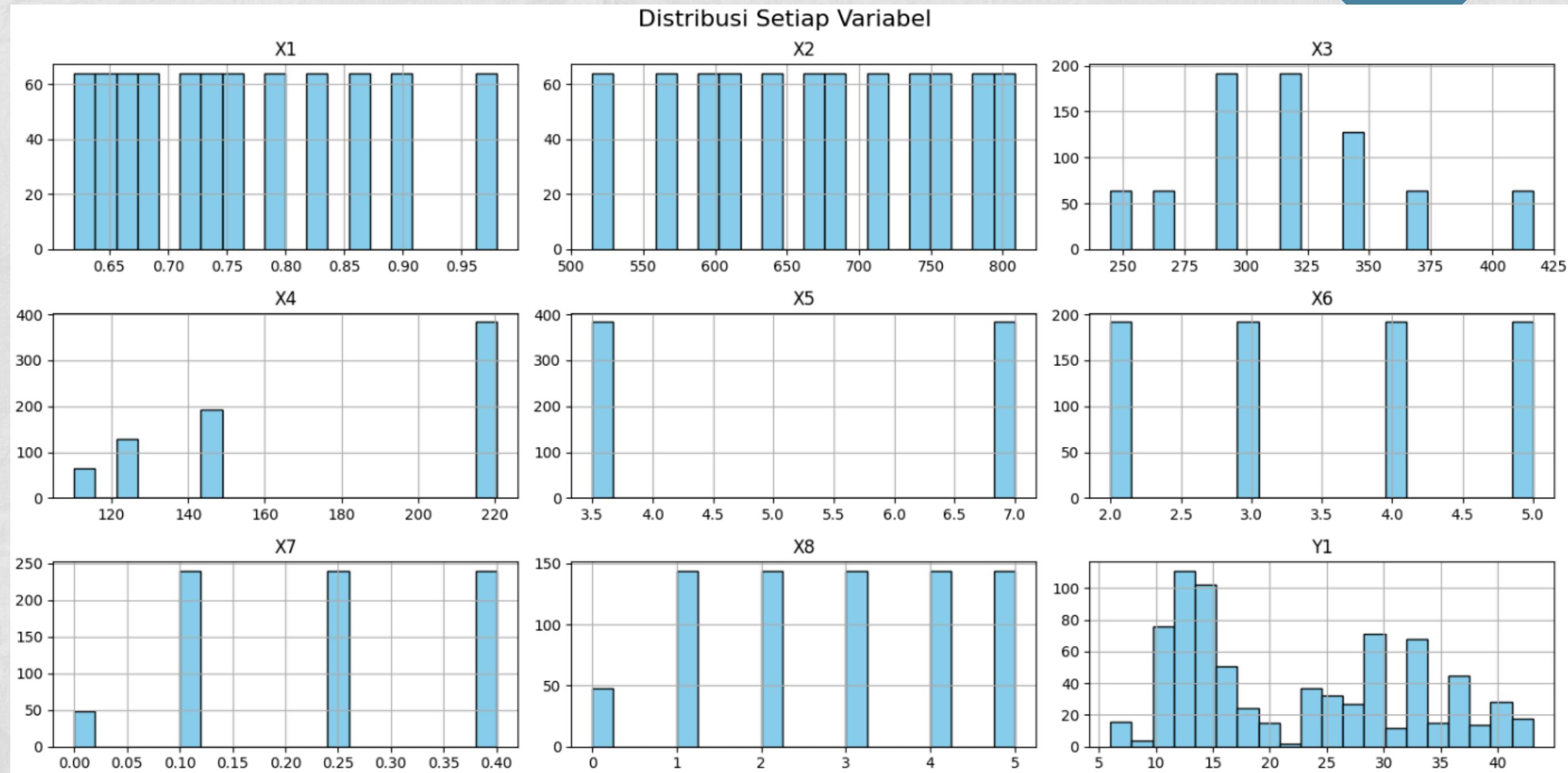
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 10 columns):
 #   Column  Non-Null Count  Dtype  
---  --     ----          ----- 
 0   X1      768 non-null    float64
 1   X2      768 non-null    float64
 2   X3      768 non-null    float64
 3   X4      768 non-null    float64
 4   X5      768 non-null    float64
 5   X6      768 non-null    int64  
 6   X7      768 non-null    float64
 7   X8      768 non-null    int64  
 8   Y1      768 non-null    float64
 9   Y2      768 non-null    float64
dtypes: float64(8), int64(2)
memory usage: 60.1 KB
```

```
# describe data
data.describe()
```

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	0.764167	671.708333	318.500000	176.604167	5.250000	3.500000	0.234375	2.81250	22.307195	24.587760
std	0.105777	88.086116	43.626481	45.165950	1.75114	1.118763	0.133221	1.55096	10.090204	9.513306
min	0.620000	514.500000	245.000000	110.250000	3.50000	2.000000	0.000000	0.00000	6.010000	10.900000
25%	0.682500	606.375000	294.000000	140.875000	3.50000	2.750000	0.100000	1.75000	12.992500	15.620000
50%	0.750000	673.750000	318.500000	183.750000	5.25000	3.500000	0.250000	3.00000	18.950000	22.080000
75%	0.830000	741.125000	343.000000	220.500000	7.00000	4.250000	0.400000	4.00000	31.667500	33.132500
max	0.980000	808.500000	416.500000	220.500000	7.00000	5.000000	0.400000	5.00000	43.100000	48.030000

4. Visualisasi Fitur Numerik



EKSPLORASI DATA DAN PRA-PEMROSESAN

Eksplorasi Data

Pada tahap eksplorasi data, dilakukan beberapa pemeriksaan awal terhadap dataset untuk memastikan kualitas data. Langkah-langkah yang dilakukan meliputi pengecekan data duplikat untuk memastikan tidak ada baris yang tercatat lebih dari satu kali, serta pengecekan missing value guna mengidentifikasi adanya nilai yang hilang. Selain itu, dilakukan perhitungan korelasi antar variabel untuk memahami hubungan antar fitur, dan pengecekan outlier untuk mendeteksi nilai-nilai ekstrem yang dapat memengaruhi analisis data. Tahapan ini penting untuk mempersiapkan data sebelum masuk ke proses pemodelan.

- 1.Cek Data Duplikat
- 2.Cek Missing Value
- 3.Cek Korelasi Data

```
# data duplikat
data.duplicated()
```

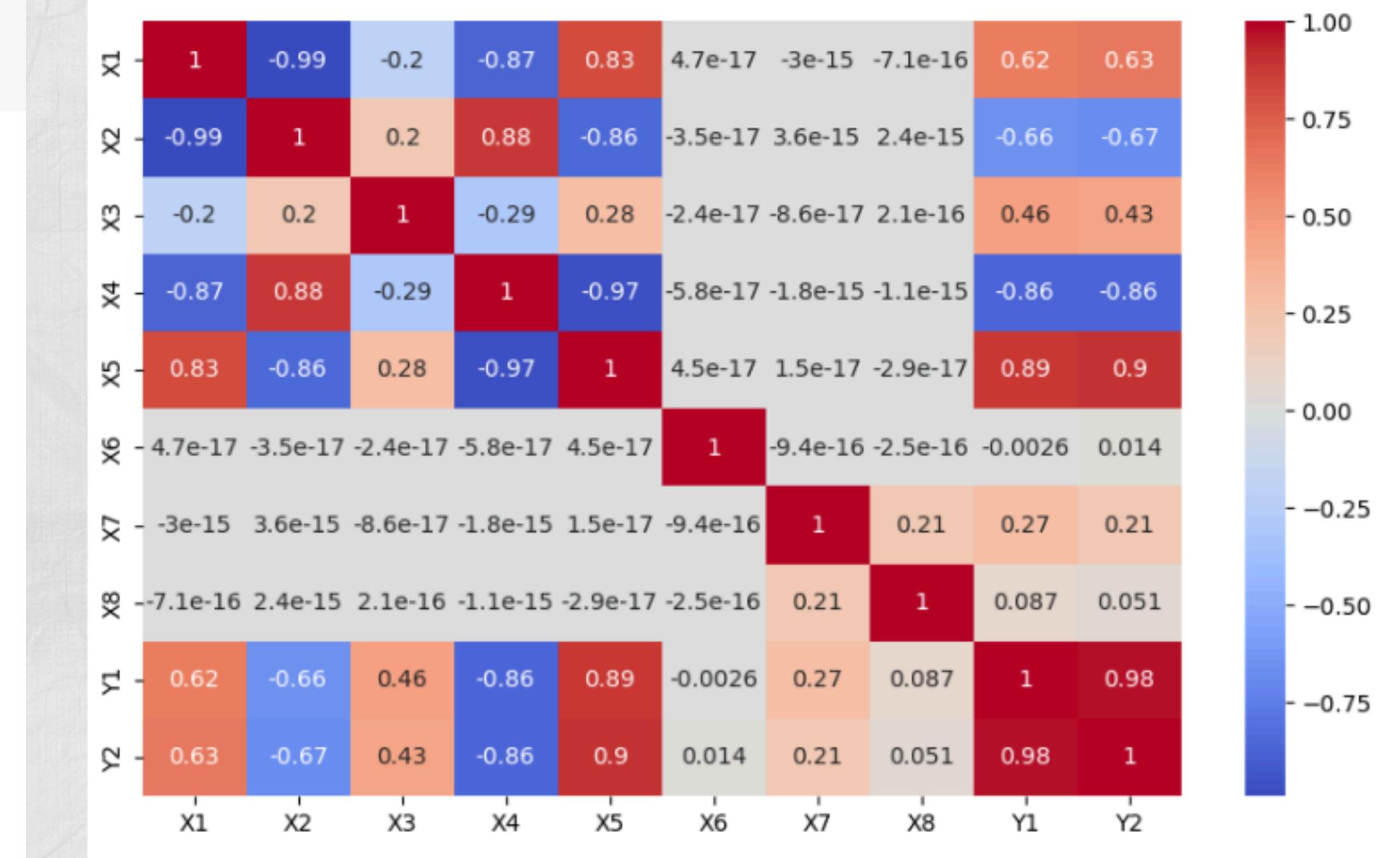
	0
0	False
1	False
2	False
3	False
4	False
...	...
763	False
764	False
765	False
766	False
767	False

768 rows × 1 columns
dtype: bool

```
# mengecek missing value
data.isna().sum()
```

	0
X1	0
X2	0
X3	0
X4	0
X5	0
X6	0
X7	0
X8	0
Y1	0
Y2	0

dtype: int64



1.Cek Outlier 2.Visualisasi Boxplot

```
def detect_outliers_iqr(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = data[(data[column] < lower_bound) | (data[column] > upper_bound)]
    return outliers

# Cek jumlah outlier per kolom
for col in num_cols:
    outliers = detect_outliers_iqr(data, col)
    print(f"Outlier pada kolom {col}: {len(outliers)} data")

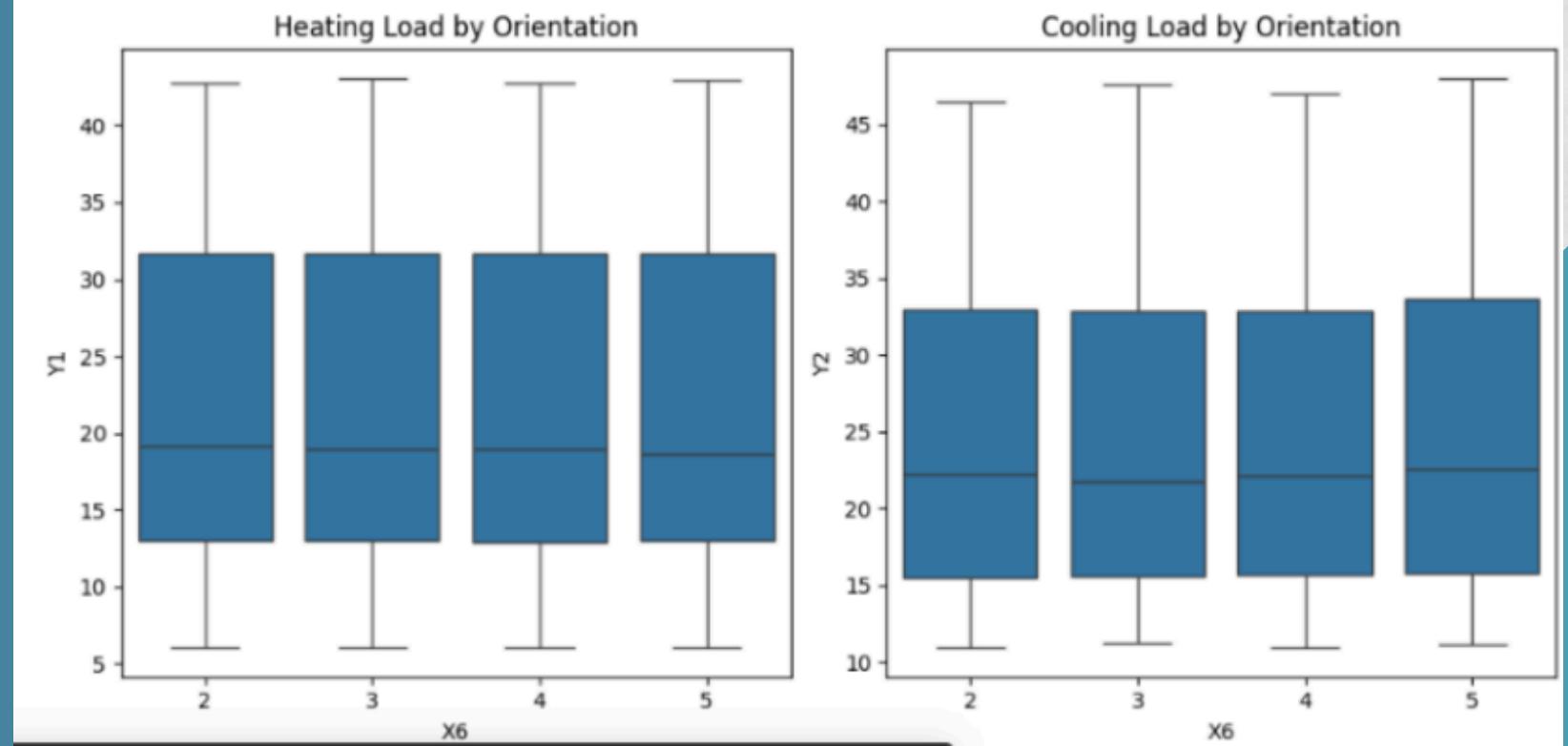
Outlier pada kolom X1: 0 data
Outlier pada kolom X2: 0 data
Outlier pada kolom X3: 0 data
Outlier pada kolom X4: 0 data
Outlier pada kolom X5: 0 data
Outlier pada kolom X7: 0 data
Outlier pada kolom Y1: 0 data
Outlier pada kolom Y2: 0 data
```

```
fig, axes = plt.subplots(1, 2, figsize=(10, 5))

# melihat apakah orientasi memiliki pengaruh pada target heating load
sns.boxplot(data=data, x='X6', y='Y1', ax=axes[0])
axes[0].set_title('Heating Load by Orientation')

# melihat apakah orientasi memiliki pengaruh pada target cooling load
sns.boxplot(data=data, x='X6', y='Y2', ax=axes[1])
axes[1].set_title('Cooling Load by Orientation')

plt.tight_layout()
plt.show()
```



EKSPLORASI DATA DAN PRA-PEMROSESAN

Pra-Pemrosesan

Pada tahap pra-pemrosesan data, dilakukan beberapa langkah penting untuk meningkatkan kualitas dan efisiensi analisis. Pertama, kolom X2 dihapus untuk menghindari kolinearitas, yaitu kondisi di mana dua variabel sangat berkorelasi dan dapat mengganggu kinerja serta interpretasi model. Selanjutnya, kolom X6 dihapus guna mengurangi dimensi dataset, karena fitur yang kurang relevan atau redundant hanya akan menambah kompleksitas tanpa memberikan kontribusi berarti. Terakhir, dilakukan transformasi pada kolom X8 (glazing_area_distribution) dengan mengubah nilai-nilai 1 hingga 5 menjadi 1, dan mempertahankan nilai 0.

Transformasi ini menghasilkan variabel biner yang lebih sederhana, namun memunculkan ketidakseimbangan data yang perlu diperhatikan dalam proses pemodelan selanjutnya.

1. Hapus Kolom X2
2. Hapus Kolom X6
3. Binarisasi

```
# drop surface_area (mencegah kolinearitas)
data.drop(['X2'], axis=1, inplace=True)
```

```
# drop Orientation (mengurangi dimensi)
data.drop(['X6'], axis=1, inplace=True)
```

```
# change glazing_area_distribution to binary variable to reduce dimensionality
data['X8'] = data['X8'].replace({0:0, 1:1, 2:1, 3:1, 4:1, 5:1})
data['X8'] = data['X8'].astype('category')
data['X8'].value_counts()
```

count

X8

1	720
0	48

IMPLEMENTASI MODEL

Pada tahap implementasi model, langkah pertama yang dilakukan adalah membagi data (split data) menjadi data latih dan data uji untuk memisahkan proses pelatihan model dengan evaluasi kinerjanya, sehingga hasil yang diperoleh lebih objektif. Selanjutnya, dilakukan pembangunan model Linear Regression, yaitu model regresi sederhana yang digunakan untuk memprediksi nilai target berdasarkan hubungan linier antara fitur dan target. Selain itu, juga dibangun model Polynomial Regression yang merupakan pengembangan dari regresi linier, dengan menambahkan derajat pangkat pada fitur agar mampu menangkap hubungan non-linier dalam data. Kedua model ini dibandingkan untuk menentukan pendekatan mana yang memberikan hasil prediksi paling akurat.

1. Split Data

```
# split the dataset into training and testing sets
X_train, X_test, y1_train, y1_test = train_test_split(x, y1, test_size = 0.30, random_state=seed)
X_train, X_test, y2_train, y2_test = train_test_split(x, y2, test_size = 0.30, random_state=seed)

# standardize the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

print("X_train shape: {}".format(X_train.shape))
print("y1_train shape: {}".format(y1_train.shape))
print("y2_train shape: {}".format(y2_train.shape))
print("X_test shape: {}".format(X_test.shape))
print("y1_test shape: {}".format(y1_test.shape))
print("y2_test shape: {}".format(y2_test.shape))

X_train shape: (537, 6)
y1_train shape: (537,)
y2_train shape: (537,)
X_test shape: (231, 6)
y1_test shape: (231,)
y2_test shape: (231,)
```

2. Model Linear Regresi

```
from sklearn.linear_model import LinearRegression

Linear = LinearRegression()

Linear_heating = Linear.fit(X_train, y1_train)
Linear_cooling = Linear.fit(X_train, y2_train)

# making predictions on the testing set
y1_pred = Linear_heating.predict(X_test)
y2_pred = Linear_cooling.predict(X_test)
```

3. Model Polynomial Regresi

```
degree = 2
poly = PolynomialFeatures(degree=degree)

X_poly_train = poly.fit_transform(X_train)
X_poly_test = poly.transform(X_test)

# Training model polynomial
poly_model_y1 = LinearRegression().fit(X_poly_train, y1_train)
poly_model_y2 = LinearRegression().fit(X_poly_train, y2_train)

# Prediksi
y1_poly_pred = poly_model_y1.predict(X_poly_test)
y2_poly_pred = poly_model_y2.predict(X_poly_test)
```

EVALUASI MODEL

1. Evaluasi Model Linear Regresi

```
# Evaluasi model untuk Target Y1 (Heating Load)
mse = mean_squared_error(y1_test, y1_pred)
mae = mean_absolute_error(y1_test, y1_pred)
r2 = r2_score(y1_test, y1_pred)

print("Target Y1 : Heating Load Model Linear")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Mean Absolute Error (MAE): {mae}")
print(f"R-squared (R2) Score: {r2}")

# Evaluasi model untuk Target Y2 (Cooling Load)
mse = mean_squared_error(y2_test, y2_pred)
mae = mean_absolute_error(y2_test, y2_pred)
r2 = r2_score(y2_test, y2_pred)

print("\nTarget Y2 : Cooling Load Model Linear")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Mean Absolute Error (MAE): {mae}")
print(f"R-squared (R2) Score: {r2}")

Target Y1 : Heating Load Model Linear
Mean Squared Error (MSE): 16.61168185621388
Mean Absolute Error (MAE): 3.391513852970394
R-squared (R2) Score: 0.8370216649634756

Target Y2 : Cooling Load Model Linear
Mean Squared Error (MSE): 12.235338342910904
Mean Absolute Error (MAE): 2.49775285370513
R-squared (R2) Score: 0.8658554982184952
```

Model regresi linear menunjukkan performa yang baik dalam memprediksi Heating Load (Y1) dan Cooling Load (Y2), dengan nilai R² masing-masing 0.837 dan 0.866. Nilai ini menunjukkan bahwa model mampu menjelaskan lebih dari 83% variasi data. Kesalahan prediksi rata-rata juga cukup rendah, yaitu 3.39 untuk Y1 dan 2.50 untuk Y2, menandakan bahwa regresi linear cukup efektif digunakan dalam memprediksi efisiensi energi bangunan pada studi ini.

2. Evaluasi Model Polynomial Regresi

```
from sklearn.preprocessing import PolynomialFeatures

# Fungsi untuk evaluasi model
def evaluate_model(y_test, y_pred, target_name=""):
    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    print(f"\nEvaluasi Model untuk {target_name}")
    print(f"Mean Squared Error (MSE): {mse:.4f}")
    print(f"Mean Absolute Error (MAE): {mae:.4f}")
    print(f"R-squared (R2) Score: {r2:.4f}")

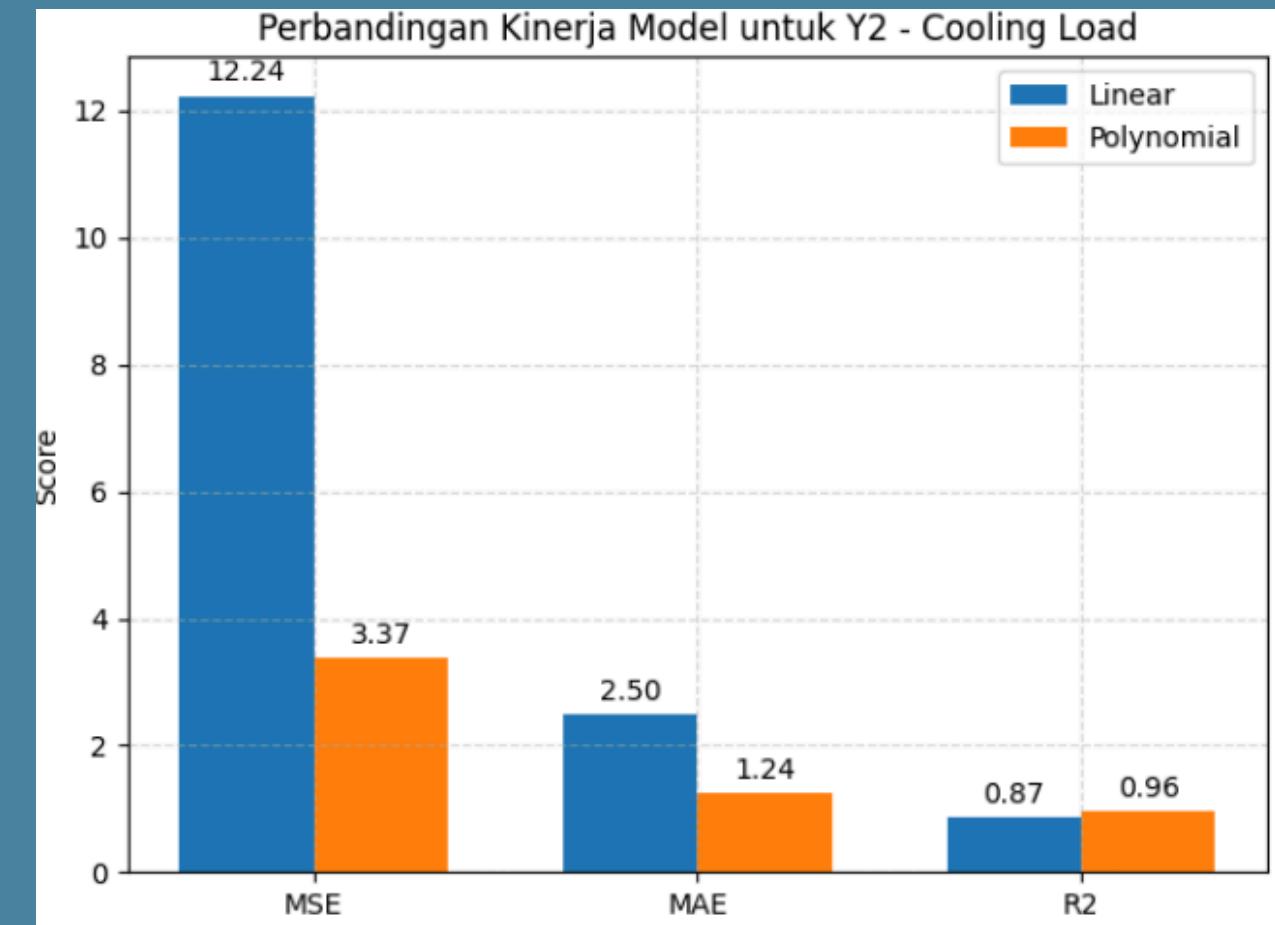
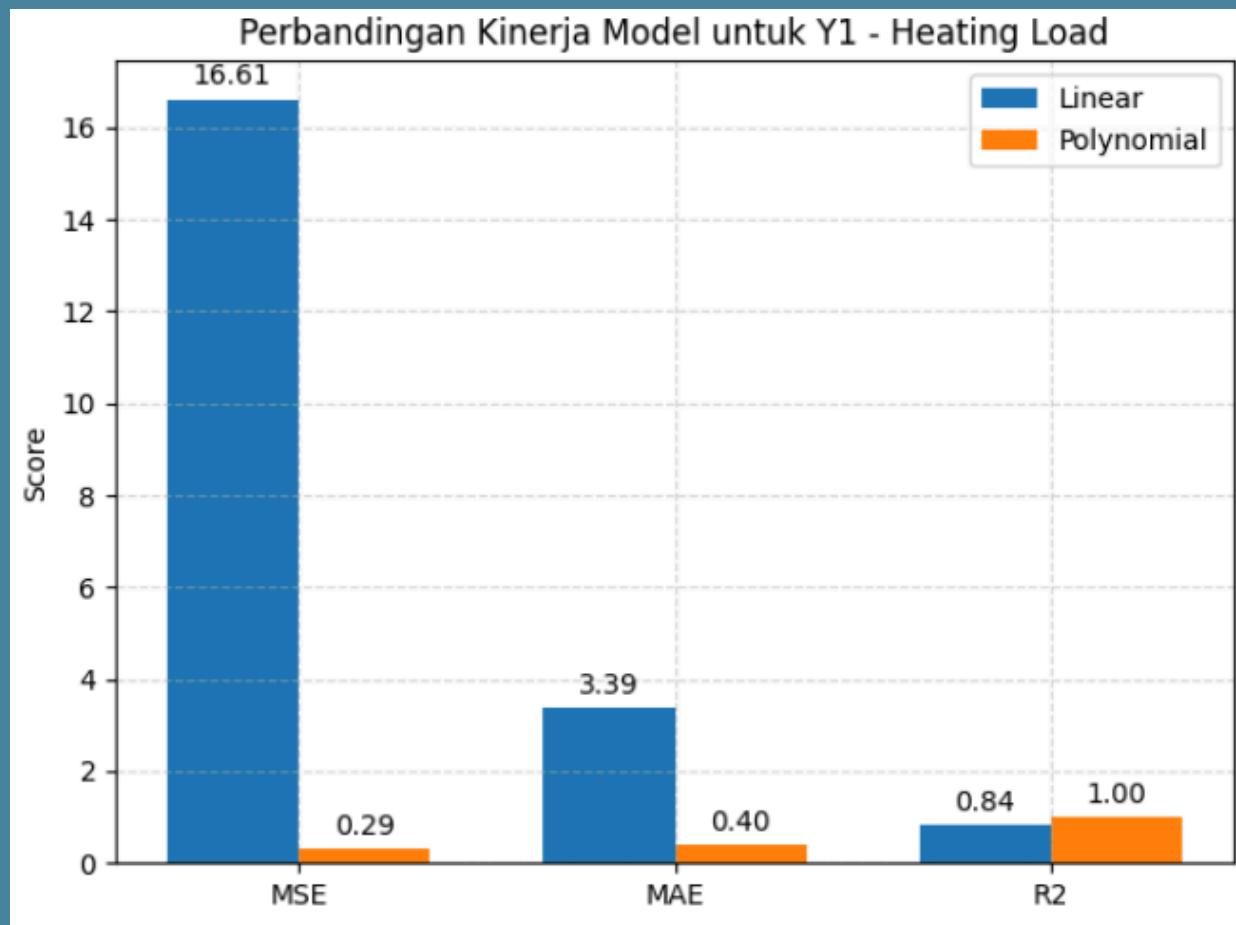
# Evaluasi Polynomial Regression
evaluate_model(y1_test, y1_poly_pred, "Y1 - Heating Load (Polynomial)")
evaluate_model(y2_test, y2_poly_pred, "Y2 - Cooling Load (Polynomial)")

Evaluasi Model untuk Y1 - Heating Load (Polynomial)
Mean Squared Error (MSE): 0.2940
Mean Absolute Error (MAE): 0.4039
R-squared (R2) Score: 0.9971

Evaluasi Model untuk Y2 - Cooling Load (Polynomial)
Mean Squared Error (MSE): 3.3726
Mean Absolute Error (MAE): 1.2419
R-squared (R2) Score: 0.9630
```

Hasil evaluasi Polynomial Regression menunjukkan performa yang sangat baik, terutama pada target Y1 (Heating Load) dengan nilai R² sebesar 0.9971 yang berarti model mampu menjelaskan 99.71% variasi pada data, serta nilai MSE dan MAE yang sangat rendah yaitu 0.2940 dan 0.4039, menandakan bahwa selisih antara prediksi dan data aktual sangat kecil. Untuk target Y2 (Cooling Load), performa model juga masih sangat baik dengan R² sebesar 0.9630, MSE sebesar 3.3726, dan MAE sebesar 1.2419, menunjukkan bahwa meskipun tidak seakurat model pada Y1, model polynomial tetap berhasil menangkap pola data dengan baik dan memiliki akurasi yang tinggi.

3. Visualisasi Perbandingan Kinerja Model



Berdasarkan hasil visualisasi perbandingan kinerja model regresi linear dan polinomial, terlihat bahwa model regresi polinomial menunjukkan performa yang jauh lebih baik dibandingkan regresi linear. Hasil ini menunjukkan bahwa hubungan antara variabel input dan output dalam dataset ini bersifat non-linear, sehingga pendekatan regresi polinomial lebih mampu menangkap pola hubungan tersebut dibandingkan model linear. Dengan kata lain, model regresi polinomial lebih sesuai digunakan untuk memprediksi Heating Load dan Cooling Load pada data ini.

HASIL ANALISIS

Linear Regresi

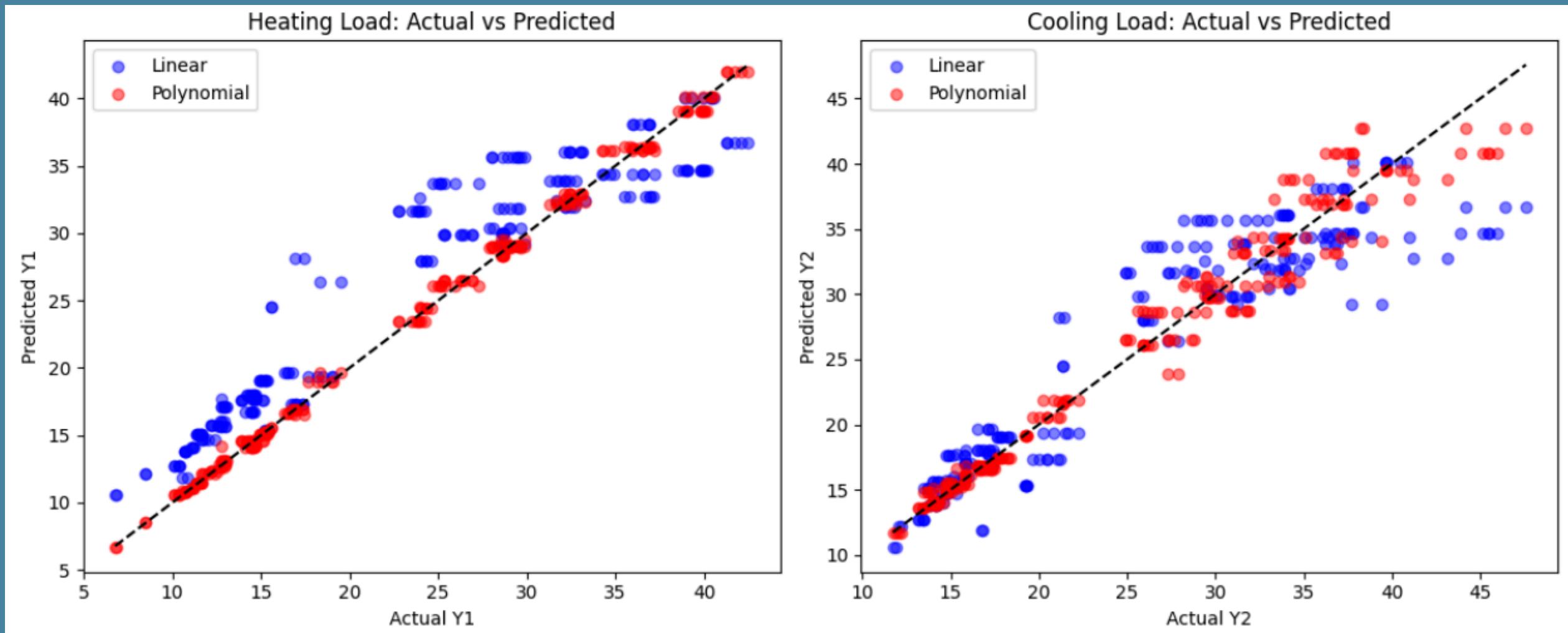
```
Koefisien Linear Regression
Intercept Model Heating Load 24.170484171322162
Koefisien Model Heating Load [-8.15611491 -2.22751434 -8.97042537 7.13172224 1.75407123 0.5130585 ]
Intercept Model Cooling Load 24.170484171322162
Koefisien Model Cooling Load [-8.15611491 -2.22751434 -8.97042537 7.13172224 1.75407123 0.5130585 ]
```

Polynomial Regresi

```
Koefisien Polynomial Regression
Intercept Model Heating Load 98.25015058779054
Koefisien Model Heating Load [ 1.47870197e-09 -9.92811641e+00 2.88136458e+01 1.09435590e+03
 1.06022377e+03 2.10608197e+00 1.13091443e-01 -2.00477840e+03
 -1.64918532e+03 -3.41968666e+03 5.02328450e+02 3.23328620e-01
 -1.71163971e-01 -3.37522692e+02 -1.32697809e+03 3.15107958e+02
 1.86280692e-01 -8.77695988e-02 4.86051421e+01 2.13164778e+03
 -9.08881193e-01 -6.00863786e-01 9.09033302e+01 -5.36138402e-01
 -1.42045293e-01 1.30190411e-01 3.29909704e-01 -4.20794649e-01]
Intercept Model Cooling Load 108.87353946336023
Koefisien Model Cooling Load [ 2.10533179e-09 -6.84503060e+01 8.06081659e+00 1.52533096e+03
 1.52374522e+03 1.67560568e+00 6.88205368e-02 -2.86369542e+03
 -2.36411185e+03 -4.87198248e+03 6.83957102e+02 6.07465282e-01
 2.06440269e-02 -4.85921184e+02 -1.89888774e+03 4.25504416e+02
 1.92752387e-01 2.09356308e-02 1.09969812e+02 3.01404007e+03
 -6.97655483e-01 3.96070570e-01 1.30645548e+02 -6.93987160e-01
 5.29648266e-01 -2.04214341e-02 2.99936392e-01 -2.56069890e-01]
```

Model regresi linear menunjukkan bahwa Relative Compactness paling berpengaruh negatif terhadap beban energi, sedangkan Roof Area berpengaruh positif. Intercept-nya sebesar 24.17. Sementara itu, model regresi polinomial memiliki koefisien lebih kompleks dan intercept lebih tinggi, namun mampu menangkap hubungan non-linear lebih baik, meski sulit diinterpretasikan.

Grafik Regression Line dan Polynomial Fit



Visualisasi scatter plot menunjukkan bahwa model Polynomial Regression (titik merah) memiliki sebaran yang lebih mendekati garis diagonal dibandingkan Linear Regression (titik biru), menandakan prediksi yang lebih akurat. Hal ini terutama terlihat jelas pada Heating Load, di mana titik Polynomial hampir menempel pada garis. Untuk Cooling Load, meskipun penyebarannya sedikit lebih lebar, Polynomial tetap menunjukkan performa yang lebih baik secara visual.

KESIMPULAN

Model yang dibuat, terutama regresi polinomial, menunjukkan performa yang sangat baik dalam memprediksi kedua target, yaitu Heating Load (Y1) dan Cooling Load (Y2). Hal ini terlihat dari nilai error yang sangat rendah dan nilai R^2 yang mendekati 1, yang menandakan bahwa model mampu menjelaskan hampir seluruh variasi data dengan sangat akurat. Sementara itu, model regresi linear memberikan hasil yang cukup baik, namun tidak seakurat model polinomial. Nilai error pada regresi linear masih tergolong tinggi, dan nilai R^2 yang lebih rendah menunjukkan bahwa model ini kurang mampu menangkap hubungan non-linear dalam data. Oleh karena itu, dapat disimpulkan bahwa meskipun model linear memberikan gambaran awal yang cukup, model polinomial jauh lebih cocok dan andal untuk prediksi pada dataset ini.



KELOMPOK 6



TERIMAKASIH