



Kelompok 1

# LINEAR DAN POLYNOMIAL

## REGRESSION



Machine Learning



# LINEAR DAN POLYNOMIAL REGRESSION

## Kelompok 1

■	<b>Aulia Vika Rahman</b>	<b>(2208107010001)</b>
■	<b>Shofia Nurul Huda</b>	<b>(2208107010015)</b>
■	<b>M. Ihsan Rizqullah Adfa</b>	<b>(2208107010029)</b>
■	<b>Nashed Bayazid Ali</b>	<b>(2208107010049)</b>
■	<b>Muhammad Raza Adzani</b>	<b>(2208107010066)</b>
■	<b>T. Farhan Syah</b>	<b>(2208107010090)</b>

# PENDAHULUAN

## LATAR BELAKANG

Dalam dunia properti, prediksi harga rumah menjadi aspek penting baik bagi pembeli, penjual, maupun pengembang. Harga rumah dipengaruhi oleh berbagai faktor seperti lokasi, luas bangunan, jumlah kamar, hingga kondisi lingkungan sosial-ekonomi. Oleh karena itu, pemodelan yang akurat sangat dibutuhkan untuk mendukung pengambilan keputusan.

## TUJUAN PROYEK

- Menerapkan dan membandingkan model Regresi Linear dan Regresi Polinomial
- Menganalisis pengaruh fitur-fitur terhadap harga rumah dan menilai model mana yang lebih akurat

## DATASET

Sumber & Deskripsi:

- Boston Housing Dataset
- Sumber: UCI Machine Learning Repository
- Jenis Data: Data real-estate (kependudukan dan perumahan) di Boston, AS
- Jumlah Sampel: 506 observasi (rumah)
- Jumlah Fitur: 13 fitur numerik + 1 label (MEDV)
- Format: CSV

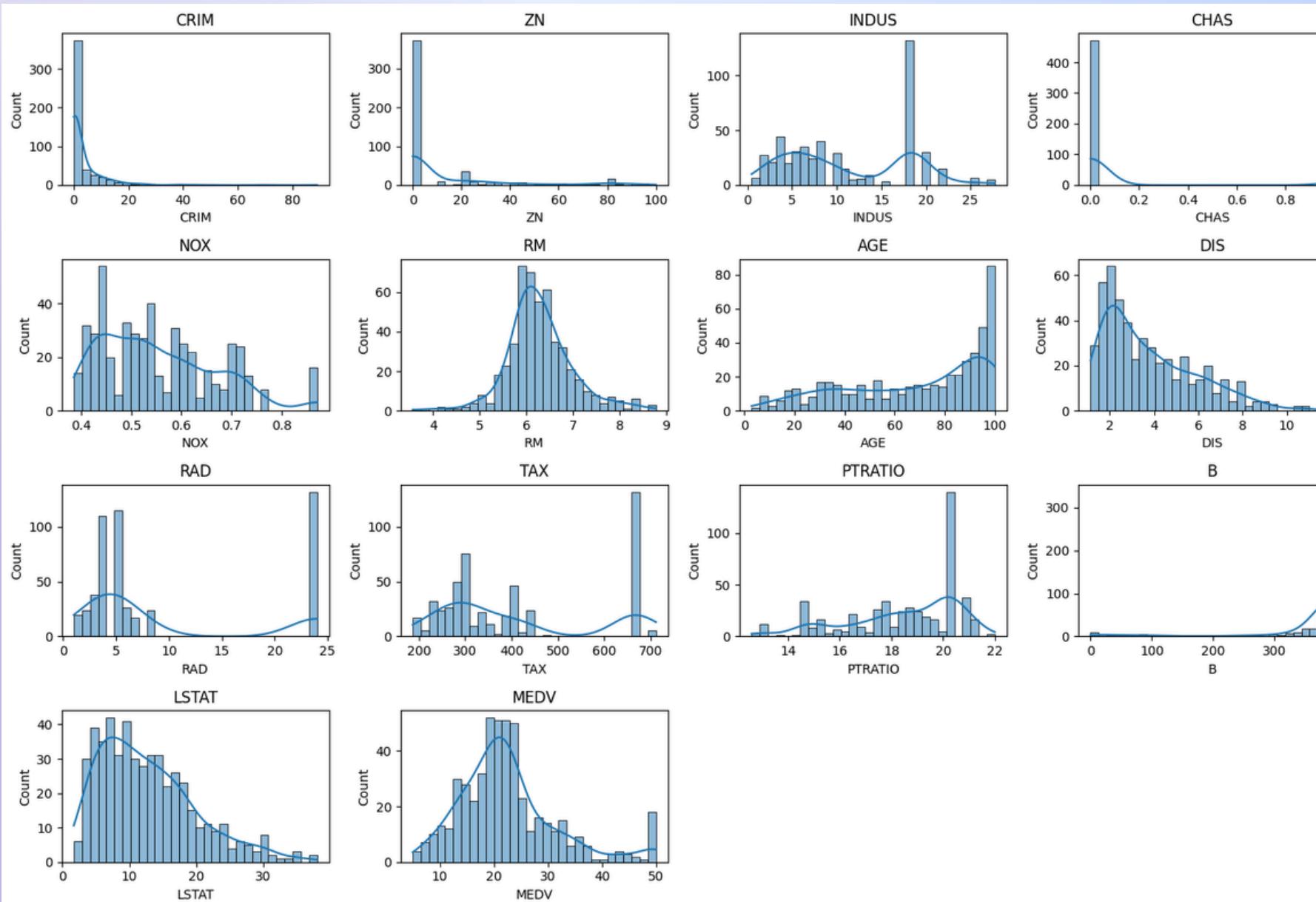
Tujuan Dataset:

- Menganalisis faktor-faktor yang memengaruhi harga rumah di Boston
- Menjadi benchmark klasik dalam eksperimen regresi (supervised learning)
- Digunakan untuk membandingkan performa berbagai algoritma prediksi harga rumah



# EKSPLORASI DATA DAN PRA-PEMROSESAN

## DISTRIBUSI NILAI FITUR



Hasil visualisasi histogram untuk setiap fitur menunjukkan:

Fitur dengan distribusi mendekati normal:

- RM (rata-rata jumlah kamar)
- MEDV (target - harga rumah)

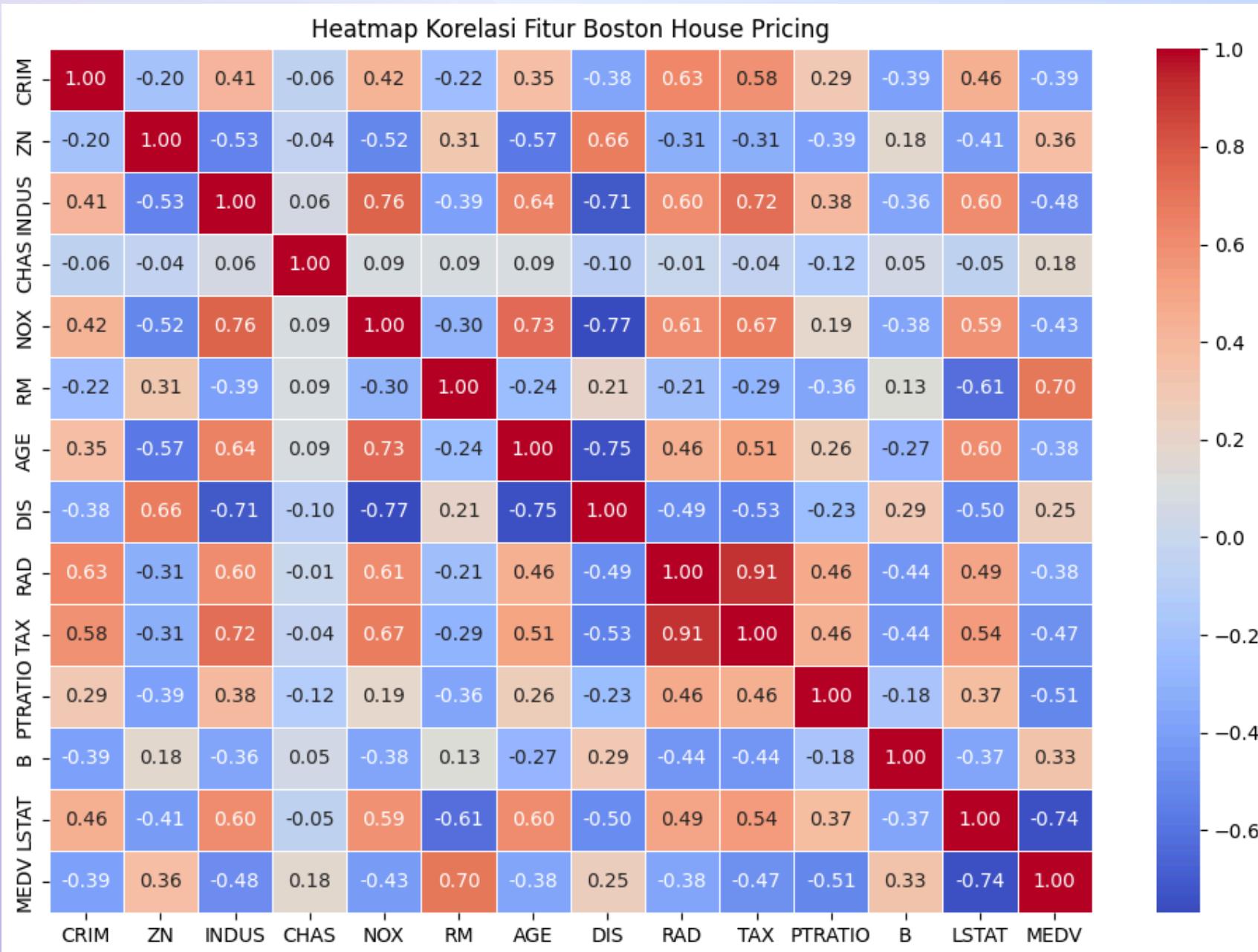
Fitur dengan distribusi skewed (miring):

- CRIM: sangat miring ke kanan (majoritas daerah memiliki tingkat kejahatan rendah)
- ZN: mayoritas nilai mendekati 0 (sedikit area dengan lahan perumahan luas)
- B: distribusi tidak merata dengan konsentrasi tinggi di nilai maksimum

Pola distribusi ini memberikan gambaran karakteristik data yang perlu diperhatikan dalam pemrosesan.

# EKSPLORASI DATA DAN PRA-PEMROSESAN

## ANALISIS KORELASI



Heatmap korelasi mengungkapkan beberapa hubungan penting:

Korelasi tinggi antar fitur:

- RAD dan TAX ( $r = 0.91$ ): aksesibilitas jalan raya berkorelasi kuat dengan tingkat pajak

Korelasi dengan target (MEDV):

- RM: korelasi positif kuat ( $r = 0.70$ ) – semakin banyak kamar, semakin tinggi harga rumah
- LSTAT: korelasi negatif kuat ( $r = -0.74$ ) – semakin tinggi persentase populasi ekonomi rendah, semakin rendah harga rumah
- INDUS, NOX, dan TAX: korelasi negatif moderat – faktor industri dan polusi menurunkan harga

Temuan ini mengarahkan langkah pra-pemrosesan untuk mengatasi multikolinearitas.

# EKSPLORASI DATA DAN PRA-PEMROSESAN

## LOG TRANSFORMATION

```
# Pilih kolom untuk log transformation
columns_to_transform = ['CRIM', 'NOX', 'TAX',
'AGE', 'DIS', 'PTRATIO', 'LSTAT', 'B']
# Lakukan log transformation
for col in columns_to_transform:
    data[col] = np.log1p(data[col]) # log(1
+ x) untuk menghindari log(0)
```

Untuk mengatasi distribusi yang skewed pada beberapa fitur, kami melakukan log transformation dengan formula  $\log(1+x)$  pada fitur berikut:

- CRIM, NOX, TAX, AGE, DIS, PTRATIO, LSTAT, dan B

Transformasi ini membantu membuat distribusi data lebih simetris dan mendekati distribusi normal.

## PEMBAGIAN DATA

```
# Pisahkan fitur (X) dan target (y)
X = data.drop(['MEDV'], axis=1) # Fitur (tanpa target)
y = data['MEDV'] # Target

# Split dataset (75% train, 25% test)
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.25, random_state=51)
```

Dataset dibagi menjadi dua bagian:

- 75% data training (380 sampel): untuk melatih model
- 25% data testing (126 sampel): untuk evaluasi model

Pembagian dilakukan secara acak untuk menghindari bias.

## STANDARISASI

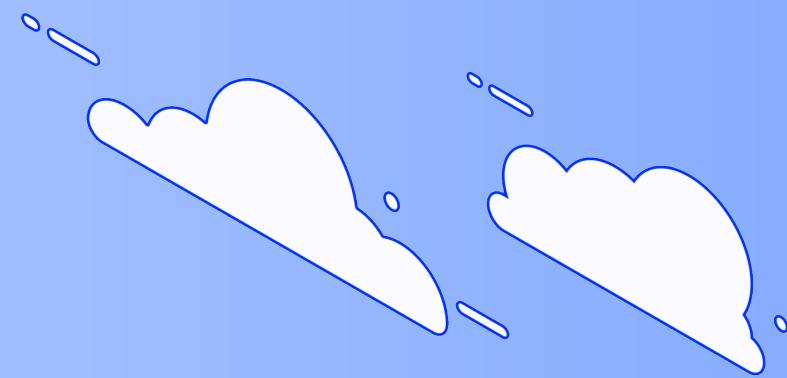
```
# Standarisasi hanya dengan data training (untuk
menghindari data leakage)**
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Kami melakukan standarisasi pada semua fitur untuk menyamakan skala data dengan:

- Mean = 0
- Standar deviasi = 1

Proses standarisasi hanya diterapkan berdasarkan data training untuk menghindari data leakage, kemudian diterapkan ke data testing.

# MODEL & EVALUASI



## IMPLEMENTASI MODEL REGRESI LINEAR



```
# **Train model Linear Regression**
model = LinearRegression()
model.fit(X_train_scaled, y_train)

# **Interpretasi Koefisien Regresi**
coef_df = pd.DataFrame(model.coef_, X.columns,
columns=['Koefisien'])
print("\n Koefisien Regresi:")
print(coef_df.sort_values(by='Koefisien',
ascending=False))
```

Model regresi linear diimplementasikan menggunakan scikit-learn:

- 1.Inisialisasi model LinearRegression
- 2.Training menggunakan data yang telah distandardisasi
- 3.Ekstraksi koefisien model untuk interpretasi

## IMPLEMENTASI MODEL REGRESI POLINOMIAL

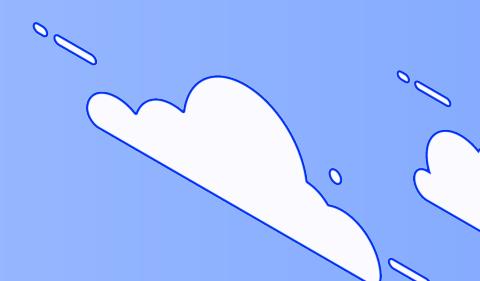


```
# Buat fitur polinomial (derajat 2)
poly = PolynomialFeatures(degree=2,
include_bias=False)
X_train_poly = poly.fit_transform(X_train)
X_test_poly = poly.transform(X_test)

# Latih model Linear Regression dengan fitur
polinomial
poly_model = LinearRegression()
poly_model.fit(X_train_poly, y_train)
```

Model polinomial derajat 2 diimplementasikan untuk menangkap pola non-linear:

- 1.Transformasi fitur menggunakan PolynomialFeatures
- 2.Menghasilkan fitur asli, kuadrat fitur, dan interaksi antar fitur
- 3.Standarisasi fitur polinomial
- 4.Training model regresi pada fitur polinomial



# MODEL & EVALUASI

## HASIL EVALUASI MODEL REGRESI LINEAR

Mean Squared Error (MSE): 20.7593  
Mean Absolute Error (MAE): 3.1947  
 $R^2$  Score: 0.7888

## HASIL EVALUASI MODEL REGRESI POLINOMIAL

$R^2$  Score - Polynomial Regression (degree=2): 0.8906  
MSE Score - Polynomial Regression (degree=2): 10.7539  
MAE Score - Polynomial Regression (degree=2): 2.4063

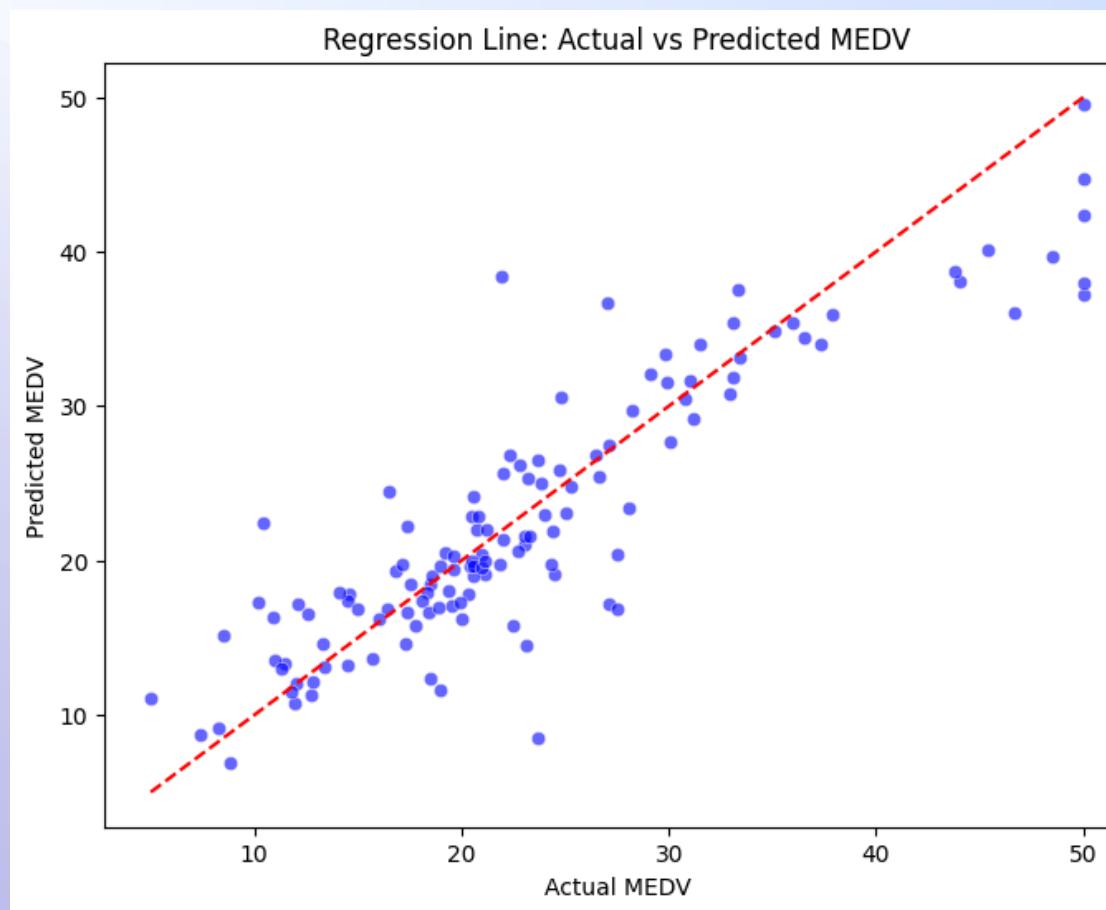
## PERBANDINGAN MODEL

Model	MSE	MAE	$R^2$ Score
Linear	20.7593	3.1947	0.7888
Polynomial	10.7539	2.4063	0.8906

Berdasarkan hasil perbandingan, model regresi polinomial (derajat 2) memberikan performa yang lebih baik dibandingkan model regresi linear pada ketiga metrik evaluasi.

# ANALISIS HASIL : VISUALISASI MODEL LINEAR VS POLYNOMIAL

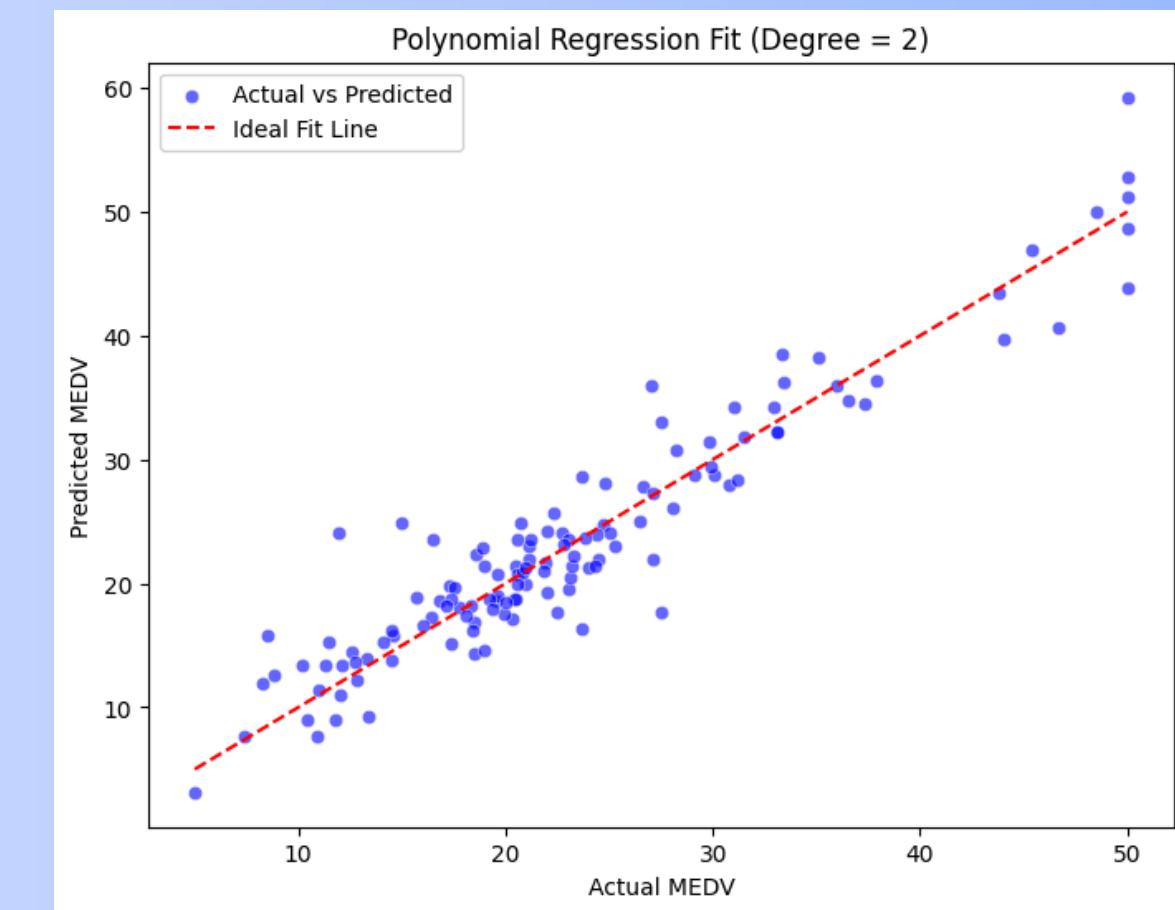
## IMPLEMENTASI MODEL REGRESI LINEAR



### Model Regresi Linear:

- Pola prediksi linear cukup mengikuti tren umum data
- Kesalahan prediksi meningkat untuk rumah harga tinggi
- Terlihat adanya pola pada residual, menandakan keterbatasan model

## IMPLEMENTASI MODEL REGRESI LINEAR



### Model Regresi Polinomial:

- Pola prediksi lebih fleksibel dan mengikuti distribusi data aktual
- Kesalahan prediksi lebih merata di seluruh rentang harga
- Lebih baik dalam menangkap kompleksitas data, terutama untuk harga ekstrem

# KESIMPULAN & SARAN

## KESIMPULAN MODEL

- Model regresi polinomial (derajat 2) secara signifikan lebih baik dibandingkan model linear
- Peningkatan  $R^2$  Score dari 0.7888 menjadi 0.8906 mengindikasikan adanya hubungan non-linear yang signifikan
- MSE turun hampir 50% pada model polinomial (dari 20.76 menjadi 10.75)

## SARAN VALIDASI

- Penerapan cross-validation untuk estimasi performa lebih robust
- Analisis model pada segmen data berbeda untuk pemahaman lebih mendalam

## SARAN PENGEMBANGAN MODEL

- Penerapan teknik regularisasi (Ridge, Lasso) untuk mengatasi overfitting
- Coba pendekatan machine learning lain (Random Forest, Gradient Boosting)

## KESIMPULAN FAKTOR HARGA

- Jumlah kamar (RM): faktor positif terkuat
- Persentase penduduk ekonomi rendah (LSTAT): faktor negatif terkuat
- Faktor lingkungan (NOX) dan aksesibilitas (DIS) juga berpengaruh signifikan

## SARAN FITUR ENGINEERING

- Penciptaan fitur baru yang relevan (rasio kamar per luas, indeks lingkungan)
- Seleksi fitur yang lebih mendalam



Kelompok 1

# THANK YOU!