Laporan Mata Kuliah Pembelajaran Mesin

Analisis Prediksi Gaji Menggunakan Regresi Linear dan Polinomial

Disusun untuk memenuhi tugas mata kuliah Pembelajaran Mesin

Oleh:

Meutia Aini	(2208107010005)
Akhsania Maisa Rahmah	(2208107010017)
Fadli Ahmad Yazid	(2208107010032)
Muhammad Mahathir	(2208107010056)
Muhammad Aufa Zaikra	(2208107010070)

JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH

2025

1 Pendahuluan

Laporan ini bertujuan untuk menganalisis dan mengimplementasikan model regresi linear serta regresi polinomial guna memprediksi gaji karyawan berdasarkan dataset "Salary Data" yang diperoleh dari Kaggle (Kaggle Salary Data).

Dataset ini berisi informasi karyawan seperti usia, jenis kelamin, tingkat pendidikan, jabatan pekerjaan, tahun pengalaman kerja, dan gaji tahunan dalam dolar. Fokus utama analisis adalah membangun model prediksi gaji menggunakan fitur utama, yaitu pengalaman kerja, tingkat pendidikan, dan jabatan pekerjaan.

Tujuan analisis ini adalah untuk:

1. Memahami karakteristik dataset melalui eksplorasi data.
2. Membangun model regresi linear dan polinomial untuk memprediksi gaji.
3. Mengevaluasi performa model menggunakan metrik seperti Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), dan R-squared (R^2).
4. Menginterpretasikan hasil model dan menentukan model terbaik.

Analisis ini dilakukan menggunakan bahasa pemrograman Python dengan library seperti `pandas`, `scikit-learn`, `matplotlib`, dan `seaborn`. Laporan ini disusun untuk memberikan gambaran yang jelas dan terstruktur mengenai proses analisis data serta hasil yang diperoleh.

2 Metodologi

2.1 Pemahaman Dataset

Dataset "Salary Data" terdiri dari 6.704 entri dengan 6 kolom utama, yaitu:

- *Age*: Usia karyawan (numerik, dalam tahun).
- *Gender*: Jenis kelamin (kategorikal: Male, Female, Other).
- *Education Level*: Tingkat pendidikan (kategorikal: High School, Bachelor's, Master's, PhD).
- *Job Title*: Jabatan pekerjaan (kategorikal, misalnya: Software Engineer, Data Analyst).
- *Years of Experience*: Tahun pengalaman kerja (numerik).
- *Salary*: Gaji tahunan dalam dolar (numerik, variabel target).

Langkah awal adalah memahami struktur dataset melalui statistik deskriptif dan visualisasi awal untuk mengeksplorasi distribusi data serta hubungan antar variabel.

2.2 Eksplorasi Data dan Pra-pemrosesan

Langkah-langkah pra-pemrosesan meliputi:

1. **Pemeriksaan Missing Values**: Mengidentifikasi dan menangani data yang hilang menggunakan fungsi `df.isna().sum()` dan `df.dropna()`.
2. **Standardisasi Nilai Kategorikal**: Memperbaiki inkonsistensi pada data kategorikal, seperti penulisan "PhD" dan "phD".
3. **Encoding Fitur Kategorikal**: Mengubah variabel kategorikal seperti *Education Level* dan *Job Title* menjadi variabel dummy menggunakan `OneHotEncoder`.
4. **Standarisasi Fitur Numerik**: Menstandarisasi fitur numerik *Years of Experience* menggunakan `StandardScaler`.
5. **Analisis Korelasi**: Membuat matriks korelasi untuk fitur numerik menggunakan `seaborn.heatmap()`.

Data kemudian dibagi menjadi 80% data latih dan 20% data uji menggunakan `train_test_split` dengan parameter `random_state=42` untuk memastikan reproduktibilitas.

2.3 Implementasi Model

Dua model utama dibangun:

- **Regresi Linear**: Menggunakan `LinearRegression` dari `scikit-learn`, dengan asumsi hubungan linier antara fitur dan gaji.
- **Regresi Polinomial**: Menggunakan `PolynomialFeatures` dengan derajat 2 dan 3 untuk menangkap hubungan non-linier, diikuti oleh `LinearRegression`.

Model dibangun menggunakan `Pipeline` untuk mengintegrasikan pra-pemrosesan dan pemodelan.

2.4 Evaluasi Model

Model dievaluasi menggunakan metrik berikut:

- **Mean Absolute Error (MAE)**: Mengukur rata-rata error absolut.
- **Mean Squared Error (MSE)**: Mengukur rata-rata kuadrat error.
- **Root Mean Squared Error (RMSE)**: Akar kuadrat dari MSE.
- **R-squared (R^2)**: Mengukur proporsi variansi yang dijelaskan oleh model.

Metrik ini dihitung menggunakan fungsi `mean_absolute_error`, `mean_squared_error`,dan `r2_score` dari `scikit-learn`.

2.5 Analisis Hasil

Hasil model divisualisasikan untuk membandingkan kecocokan regresi linear dan polinomial terhadap data aktual. Koefisien model juga diinterpretasikan untuk memahami pengaruh masing-masing fitur terhadap gaji.

3 Hasil dan Proses

3.1 Pemahaman Dataset

Informasi dataset diperoleh dengan kode berikut:

```
import pandas as pd
df = pd.read_csv('Salary_Data.csv')
df.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6704 entries, 0 to 6703
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Age                  6702 non-null   float64
 1   Gender               6702 non-null   object
 2   Education Level      6701 non-null   object
 3   Job Title            6702 non-null   object
 4   Years of Experience  6701 non-null   float64
 5   Salary               6699 non-null   float64
dtypes: float64(3), object(3)
memory usage: 314.4+ KB
```

Kolom	Jumlah Non-Null	Tipe Data
Age	6702	float64
Gender	6702	object
Education Level	6701	object
Job Title	6702	object
Years of Experience	6701	float64
Salary	6699	float64

Table 1: Informasi Dataset "Salary Data"

Statistik deskriptif untuk fitur numerik:

```
print(df.describe())
```

Output:

```
                Age  Years of Experience         Salary
count   6702.000000           6701.000000    6699.000000
mean      33.620859              8.094687  115326.964771
std        7.614633              6.059003   52786.183911
min       21.000000              0.000000     350.000000
25%       28.000000              3.000000   70000.000000
50%       32.000000              7.000000  115000.000000
75%       38.000000             12.000000  160000.000000
max       62.000000             34.000000  250000.000000
```

Statistik	Age	Years of Experience	Salary
Count	6702	6701	6699
Mean	33.62	8.09	115326.96
Std	7.61	6.06	52786.18
Min	21.00	0.00	350.00
25%	28.00	3.00	70000.00
50%	32.00	7.00	115000.00
75%	38.00	12.00	160000.00
Max	62.00	34.00	250000.00

Table 2: Statistik Deskriptif Fitur Numerik

Contoh data awal:

Index	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

Table 3: Contoh Data Awal

3.2 Eksplorasi Data

3.2.1 Distribusi Fitur Numerik dan Kategorikal

Kode visualisasi untuk distribusi numerik dan kategorikal (digabung menjadi satu):

```
import matplotlib.pyplot as plt
import seaborn as sns

```

```
fig, axes = plt.subplots(2, 3, figsize=(18, 10))

# Distribusi Usia
sns.histplot(df['Age'], kde=True, ax=axes[0, 0], color='blue')
axes[0, 0].set_title('Distribusi Usia')

# Distribusi Pengalaman Kerja
sns.histplot(df['Years of Experience'], kde=True, ax=axes[0,
    1], color='green')
axes[0, 1].set_title('Distribusi Pengalaman Kerja')

# Distribusi Gaji
sns.histplot(df['Salary'], kde=True, ax=axes[0, 2], color='red')
axes[0, 2].set_title('Distribusi Gaji')

# Distribusi Jenis Kelamin
gender_counts = df['Gender'].value_counts()
axes[1, 0].pie(gender_counts, labels=gender_counts.index,
    autopct='%1.1f%%')
axes[1, 0].set_title('Distribusi Jenis Kelamin')

# Distribusi Tingkat Pendidikan
edu_counts = df['Education Level'].value_counts()
sns.barplot(x=edu_counts.index, y=edu_counts.values, ax=axes[1,
    1])
axes[1, 1].set_title('Distribusi Tingkat Pendidikan')
axes[1, 1].set_xticklabels(axes[1, 1].get_xticklabels(),
    rotation=45, ha='right')

# Distribusi Jabatan (Top 5)
job_counts = df['Job Title'].value_counts().head(5)
sns.barplot(x=job_counts.index, y=job_counts.values, ax=axes[1,
    2])
axes[1, 2].set_title('Top 5 Jabatan')
axes[1, 2].set_xticklabels(axes[1, 2].get_xticklabels(),
    rotation=45, ha='right')

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

Interpretasi: Visualisasi ini menampilkan distribusi fitur numerik dan kategorikal dalam dataset. Untuk fitur numerik:

- *Distribusi Usia*: Usia karyawan berkisar antara 21 hingga 62 tahun, dengan mayoritas berada di rentang 28–38 tahun.
- *Distribusi Pengalaman Kerja*: Pengalaman kerja berkisar dari 0 hingga 34 tahun, dengan mayoritas karyawan memiliki pengalaman 3–12 tahun.
- *Distribusi Gaji*: Gaji berkisar dari $350 hingga $250.000, dengan distribusi mul-

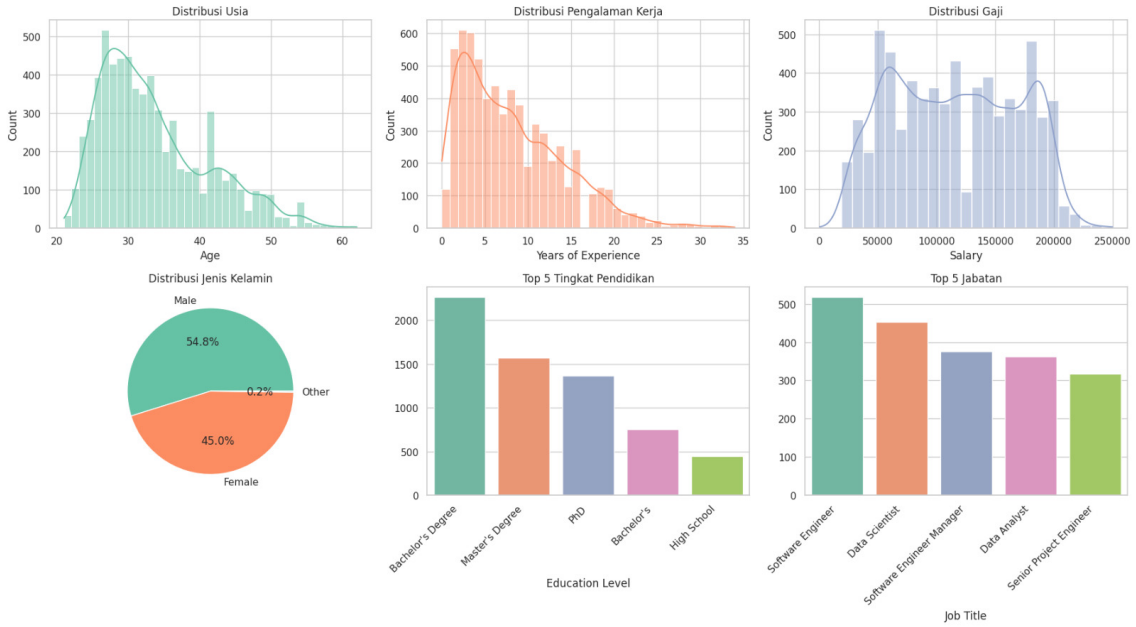

Figure 1: Distribusi Fitur Numerik dan Kategorikal

timodal yang menunjukkan puncak di sekitar $50.000, $100.000, dan $150.000–$200.000.

Untuk fitur kategorikal:

- *Distribusi Jenis Kelamin*: Distribusi hampir seimbang dengan 54.8% Male, 45.0% Female, dan 0.2% Other.
- *Tingkat Pendidikan*: Mayoritas karyawan memiliki gelar Bachelor's, diikuti oleh Master's, PhD, dan High School.
- *Jabatan*: Jabatan teratas meliputi Software Engineer, Data Scientist, Senior Manager, Data Analyst, dan Project Engineer.

3.2.2 Hubungan Antar Variabel

Kode visualisasi:

```
fig, axes = plt.subplots(2, 3, figsize=(18, 10))

# Usia vs Salary
sns.scatterplot(x=df['Age'], y=df['Salary'], ax=axes[0, 0],
    color='red')
axes[0, 0].set_title('Hubungan Usia dengan Gaji')

# Pengalaman vs Salary
```

```
sns.scatterplot(x=df['Years of Experience'], y=df['Salary'],
    ax=axes[0, 1], color='yellow')
axes[0, 1].set_title('Hubungan Pengalaman Kerja dengan Gaji')

# Gender vs Salary
sns.boxplot(x=df['Gender'], y=df['Salary'], ax=axes[0, 2])
axes[0, 2].set_title('Distribusi Gaji berdasarkan Gender')

# Pendidikan vs Salary
sns.boxplot(x=df['Education Level'], y=df['Salary'], ax=axes[1,
    0])
axes[1, 0].set_title('Distribusi Gaji berdasarkan Pendidikan')
axes[1, 0].set_xticklabels(axes[1, 0].get_xticklabels(),
    rotation=45, ha='right')

# Rata-rata gaji berdasarkan Job Title (Top 10)
top_jobs = df['Job Title'].value_counts().nlargest(10).index
df['Job Title (Filtered)'] = df['Job Title'].apply(lambda x: x
    if x in top_jobs else 'Other')
job_salary_avg = df.groupby('Job Title
    (Filtered)')['Salary'].mean().sort_values()
sns.barplot(x=job_salary_avg.index, y=job_salary_avg.values,
    ax=axes[1, 1])
axes[1, 1].set_title('Rata-rata Gaji berdasarkan Job Title (Top
    10)')
axes[1, 1].set_xticklabels(axes[1, 1].get_xticklabels(),
    rotation=45, ha='right')

fig.delaxes(axes[1, 2])
plt.tight_layout()
plt.show()
```

Interpretasi:

- *Usia vs Gaji*: Terdapat hubungan positif antara usia dan gaji, meskipun variasinya cukup besar.
- *Pengalaman Kerja vs Gaji*: Hubungan non-linier yang jelas terlihat, dengan peningkatan gaji yang lebih cepat pada tahun-tahun awal pengalaman.
- *Gender vs Gaji*: Distribusi gaji antara Male dan Female relatif seimbang, dengan sedikit variasi.
- *Pendidikan vs Gaji*: Gaji meningkat seiring dengan tingkat pendidikan, dengan PhD memiliki median gaji tertinggi.
- *Jabatan vs Gaji*: Jabatan seperti Director dan Senior Manager memiliki rata-rata gaji tertinggi.

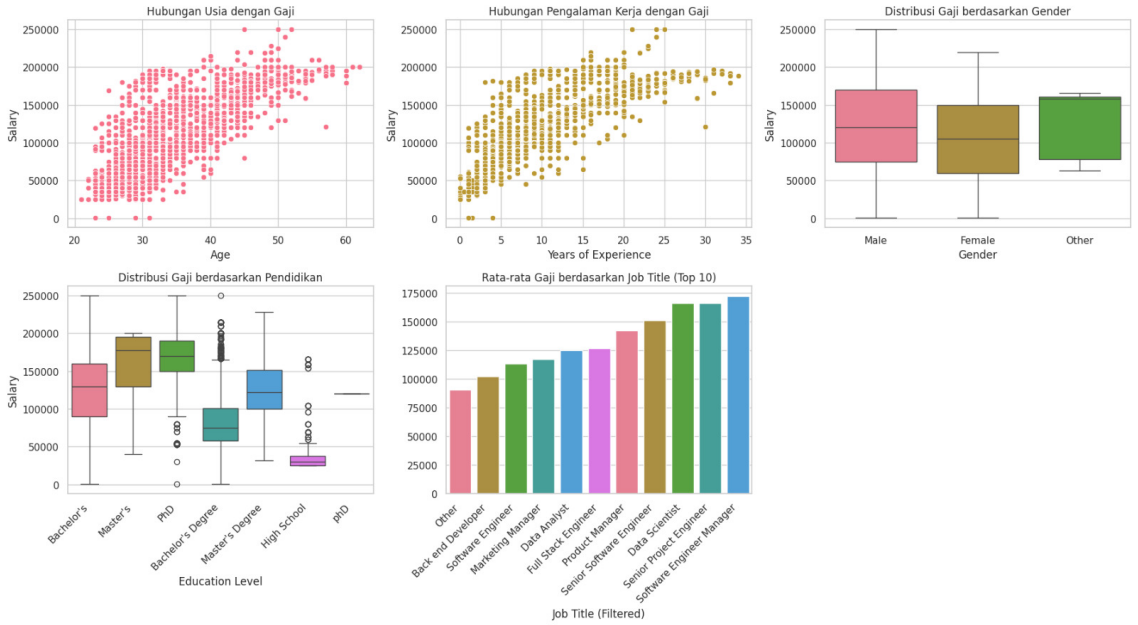

Figure 2: Hubungan Antar Variabel: Usia vs Gaji, Pengalaman Kerja vs Gaji, Gender vs Gaji, Pendidikan vs Gaji, dan Rata-rata Gaji berdasarkan Jabatan

3.2.3 Analisis Korelasi

Kode:

```
plt.figure(figsize=(8, 6))
sns.heatmap(df[['Age', 'Years of Experience',
    'Salary']].corr(), annot=True, cmap='coolwarm')
plt.title('Korelasi antar Variabel')
plt.show()
```

Interpretasi: Korelasi antara *Years of Experience* dan *Salary* adalah 0.81, menunjukkan hubungan positif yang kuat. *Age* juga memiliki korelasi positif dengan *Salary* (0.73), tetapi lebih lemah dibandingkan pengalaman kerja.

3.3 Pra-pemrosesan Data

3.3.1 Pemeriksaan Missing Values

Kode:

```
# Menghapus baris dengan nilai yang hilang
df = df.dropna()

# Memeriksa nilai yang hilang setelah penghapusan
print("Nilai yang hilang setelah penghapusan:")
print(df.isnull().sum())
```

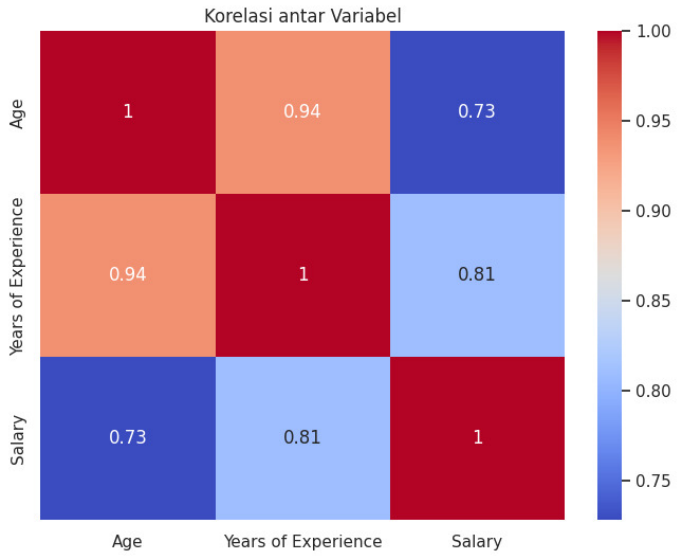

Figure 3: Matriks Korelasi antara Variabel Numerik

Output awal:

```
Age                    2
Gender                 2
Education Level        3
Job Title              2
Years of Experience    3
Salary                 5
dtype: int64
```

Output setelah penghapusan:

```
Age                    0
Gender                 0
Education Level        0
Job Title              0
Years of Experience    0
Salary                 0
dtype: int64
```

Alasan Penghapusan: Jumlah missing values sangat kecil (<1% dari total 6.704 entri), sehingga penghapusan tidak signifikan memengaruhi representativitas dataset.

3.3.2 Standardisasi Nilai Kategorikal

Kode:

```
print("\nNilai unik pada kolom Education Level:")
print(df['Education Level'].unique())
```

```
# Standardisasi Education Level
education_mapping = {
    "Bachelor's": "Bachelor's Degree",
    "Master's": "Master's Degree",
    "phD": "PhD"
}
df['Education Level'] = df['Education
   Level'].replace(education_mapping)

# Memeriksa hasil standardisasi
print("\nSetelah standardisasi Education Level:")
print(df['Education Level'].unique())
```

Output:

```
Nilai unik pada kolom Education Level:
["Bachelor's" "Master's" 'PhD' 'High School' "Bachelor's Degree" "Maste

Setelah standardisasi Education Level:
['Bachelor's Degree' "Master's Degree" 'PhD' 'High School']
```

Interpretasi: Standardisasi dilakukan untuk menghilangkan inkonsistensi penulisan seperti "PhD" dan "phD", sehingga data lebih seragam.

3.3.3 Persiapan Data untuk Pemodelan

Kode:

```
# Membuat dataset hanya dengan 3 fitur yang diminta
model_df = df[['Years of Experience', 'Education Level', 'Job
   Title', 'Salary']]

# Memisahkan fitur dan target
X = model_df.drop('Salary', axis=1)
y = model_df['Salary']

# Mendefinisikan kolom numerik dan kategorikal
numerical_cols = ['Years of Experience']
categorical_cols = ['Education Level', 'Job Title']

# Membuat preprocessor dengan ColumnTransformer
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import StandardScaler, OneHotEncoder

preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_cols),
```

```
        ('cat', OneHotEncoder(handle_unknown='ignore'),
           categorical_cols)
    ])

# Split data menjadi training dan testing
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)

print("\nUkuran data setelah splitting:")
print(f"Training set: {X_train.shape} sampel")
print(f"Test set: {X_test.shape} sampel")
```

Output:

```
Ukuran data setelah splitting:
Training set: (5352, 3) sampel
Test set: (1338, 3) sampel
```

3.4 Implementasi Model

3.4.1 Regresi Linear

Kode:

```
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression

# Linear Regression Pipeline
lr_pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', LinearRegression())
])

# Training Linear Regression
lr_pipeline.fit(X_train, y_train)
y_pred_lr = lr_pipeline.predict(X_test)
```

3.4.2 Regresi Polinomial

Kode:

```
from sklearn.preprocessing import PolynomialFeatures

# Polynomial Regression Pipelines
poly_degrees = [2, 3]
poly_predictions = {}
```

```
6
7  for degree in poly_degrees:
8      poly_pipeline = Pipeline([
9          ('preprocessor', preprocessor),
10         ('poly_features', PolynomialFeatures(degree=degree)),
11         ('regressor', LinearRegression())
12     ])
13     poly_pipeline.fit(X_train, y_train)
14     poly_predictions[degree] = poly_pipeline.predict(X_test)
```

3.5 Evaluasi Model

3.5.1 Evaluasi Regresi Linear

Kode:

```
1  from sklearn.metrics import mean_absolute_error,
       mean_squared_error, r2_score
2  import numpy as np
3
4  # Evaluasi Linear Regression
5  lr_metrics = {
6      'MAE': mean_absolute_error(y_test, y_pred_lr),
7      'MSE': mean_squared_error(y_test, y_pred_lr),
8      'RMSE': np.sqrt(mean_squared_error(y_test, y_pred_lr)),
9      'R2': r2_score(y_test, y_pred_lr)
10 }
11
12 print("Linear Regression Performance:")
13 for metric, value in lr_metrics.items():
14     print(f"{metric}: {value:.2f}")
```

Output:

```
Linear Regression Performance:
MAE: 15487.83
MSE: 458246826.30
RMSE: 21406.70
R2: 0.84
```

3.5.2 Evaluasi Regresi Polinomial

Kode:

```
1  poly_results = {}
2
3  for degree, y_pred_poly in poly_predictions.items():
```

```
    poly_results[degree] = {
        'MAE': mean_absolute_error(y_test, y_pred_poly),
        'MSE': mean_squared_error(y_test, y_pred_poly),
        'RMSE': np.sqrt(mean_squared_error(y_test,
            y_pred_poly)),
        'R2': r2_score(y_test, y_pred_poly)
    }
    print(f"\nPolynomial Regression (Degree {degree})
        Performance:")
    for metric, value in poly_results[degree].items():
        print(f"{metric}: {value:.2f}")
```

Output:

```
Polynomial Regression (Degree 2) Performance:
MAE: 10155.95
MSE: 247653840.17
RMSE: 15737.02
R2: 0.91

Polynomial Regression (Degree 3) Performance:
MAE: 9940.47
MSE: 740203043.98
RMSE: 27206.67
R2: 0.74
```

Hasil evaluasi model disajikan dalam tabel berikut:

Model	MAE	MSE	RMSE	R^2
Regresi Linear	15487.83	458246826.30	21406.70	0.84
Regresi Polinomial (Derajat 2)	10155.95	247653840.17	15737.02	0.91
Regresi Polinomial (Derajat 3)	9940.47	740203043.98	27206.67	0.74

Table 4: Perbandingan Performa Model

Interpretasi:

- *Regresi Linear*: R^2 sebesar 0.84 menunjukkan model cukup baik, tetapi MSE dan MAE yang tinggi mengindikasikan adanya error besar pada beberapa prediksi.
- *Regresi Polinomial Derajat 2*: R^2 tertinggi (0.91) dan MSE serta MAE terendah menunjukkan performa terbaik.
- *Regresi Polinomial Derajat 3*: R^2 menurun (0.74) dan MSE meningkat, menunjukkan tanda-tanda *overfitting*.

3.5.3 Visualisasi Performa Model

Kode visualisasi:

```
metrics_df = pd.DataFrame({
    'Linear': lr_metrics,
    'Poly2': poly_results[2]
}).T

fig, axes = plt.subplots(2, 2, figsize=(15, 10))
for i, metric in enumerate(['MAE', 'MSE', 'RMSE', 'R2']):
    ax = axes[i//2, i%2]
    metrics_df[metric].plot(kind='bar', ax=ax)
    ax.set_title(metric)
    ax.set_ylabel('Value' if metric != 'R2' else 'Score')
plt.tight_layout()
plt.show()
```

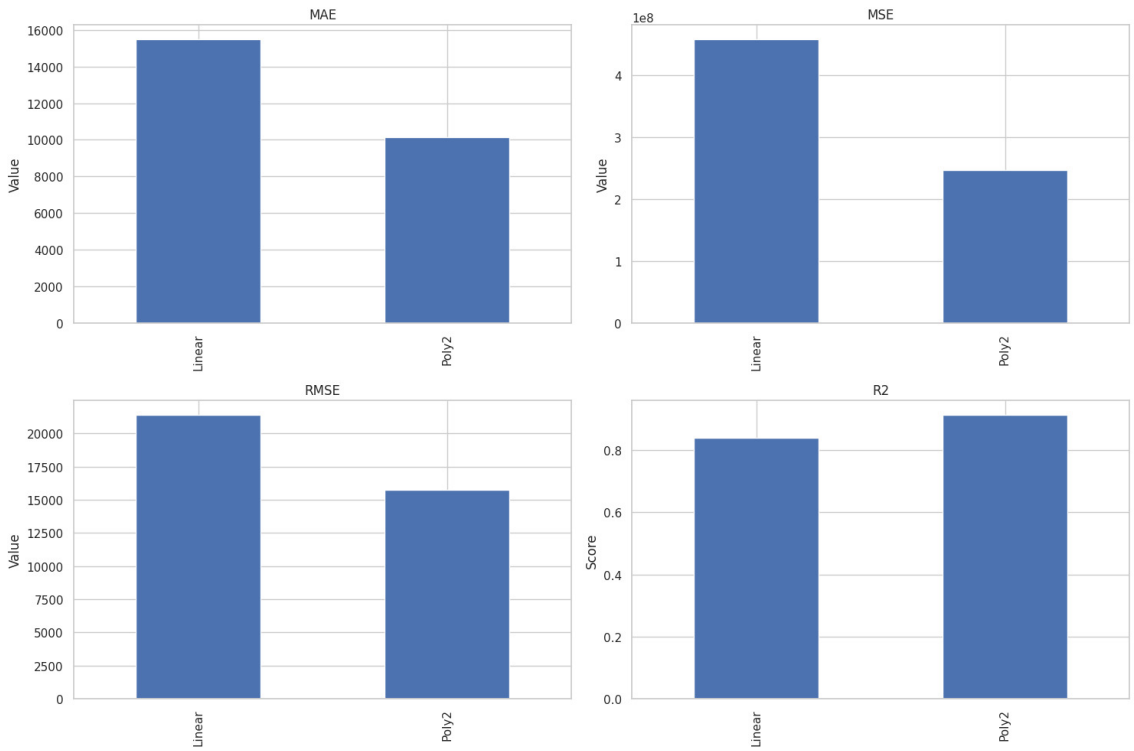

Figure 4: Perbandingan Metrik Evaluasi Model: MAE, MSE, RMSE, dan R^2

Interpretasi: Visualisasi menunjukkan bahwa regresi polinomial derajat 2 secara konsisten memiliki nilai MAE, MSE, dan RMSE yang lebih rendah, serta R^2 yang lebih tinggi dibandingkan regresi linear.

3.6 Analisis Hasil

3.6.1 Visualisasi Regresi

Kode:

```
def visualisasi_model_diperbaiki(X_test, y_test, lr_pipeline,
    poly_pipeline):
     x_exp = X_test['Years of Experience'].values.reshape(-1, 1)
     x_range = np.linspace(X_test['Years of Experience'].min(),
                           X_test['Years of Experience'].max(),
                           100).reshape(-1, 1)
     X_simple_range = pd.DataFrame(x_range, columns=['Years of
        Experience'])
     for col in X_test.columns:
         if col != 'Years of Experience':
             most_common = X_test[col].mode()[0]
             X_simple_range[col] = most_common
     y_pred_lr_line = lr_pipeline.predict(X_simple_range)
     y_pred_poly_line = poly_pipeline.predict(X_simple_range)
     plt.figure(figsize=(14, 8))
     plt.scatter(X_test['Years of Experience'], y_test,
        color='blue', alpha=0.6, label='Data Aktual')
     plt.plot(x_range, y_pred_lr_line, color='red', linewidth=2,
        label='Linear Regression (R  = 0.84)')
     plt.plot(x_range, y_pred_poly_line, color='green',
        linewidth=2, label='Polynomial Regression (R  = 0.91)')
     plt.title('Perbandingan Model Regresi Linear dan Polynomial
        pada Prediksi Gaji')
     plt.xlabel('Tahun Pengalaman')
     plt.ylabel('Gaji')
     plt.legend()
     plt.grid(True, alpha=0.3)
     plt.tight_layout()
     plt.show()

visualisasi_model_diperbaiki(X_test, y_test, lr_pipeline,
    poly_pipeline)
```

Interpretasi:

- *Regresi Linear*: Garis regresi linear menunjukkan hubungan positif yang kuat antara pengalaman kerja dan gaji, dengan $R^2 = 0.84$.
- *Regresi Polinomial*: Kurva polinomial derajat 2 lebih sesuai dengan data, dengan $R^2 = 0.91$, menangkap pola non-linier seperti perlambatan kenaikan gaji pada pengalaman tinggi.

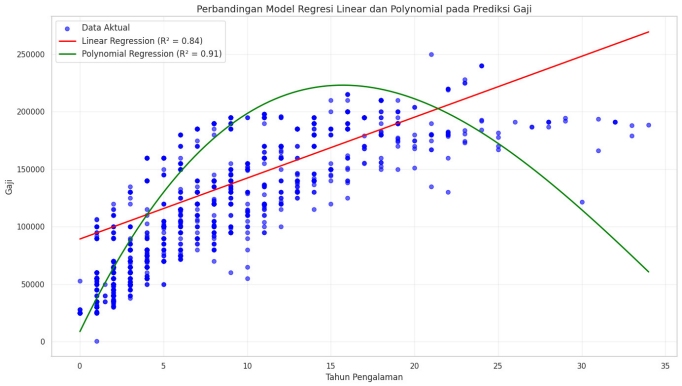

Figure 5: Perbandingan Model Regresi Linear dan Polynomial

4 Diskusi

Berdasarkan hasil evaluasi, regresi polinomial derajat 2 menunjukkan performa terbaik dengan R^2 sebesar 0.91, MSE sebesar 247653840.17, RMSE sebesar 15737.02, dan MAE sebesar 10155.95. Model ini lebih unggul dibandingkan regresi linear (R^2: 0.84, MSE: 458246826.30, RMSE: 21406.70, MAE: 15487.83) karena mampu menangkap hubungan non-linier antara pengalaman kerja dan gaji. Regresi polinomial derajat 3, meskipun memiliki MAE yang sedikit lebih rendah (9940.47), menunjukkan tanda-tanda *overfitting* dengan R^2 yang lebih rendah (0.74) dan MSE yang jauh lebih tinggi (740203043.98).

Visualisasi regresi mengkonfirmasi bahwa hubungan antara pengalaman kerja dan gaji memiliki komponen non-linier, yang diakomodasi lebih baik oleh model polinomial. Namun, model ini perlu diuji lebih lanjut pada data baru untuk memastikan generalisasi yang baik.

5 Kesimpulan

Analisis ini berhasil mengimplementasikan dan membandingkan model regresi linear serta regresi polinomial untuk memprediksi gaji karyawan berdasarkan dataset "Salary Data". Hasil utama adalah sebagai berikut:

- **Regresi Linear**: R^2 = 0.84, MSE = 458246826.30, RMSE = 21406.70, MAE = 15487.83.
- **Regresi Polinomial Derajat 2**: R^2 = 0.91, MSE = 247653840.17, RMSE = 15737.02, MAE = 10155.95.
- **Regresi Polinomial Derajat 3**: R^2 = 0.74, MSE = 740203043.98, RMSE = 27206.67, MAE = 9940.47.

Regresi polinomial derajat 2 terbukti sebagai model terbaik karena memberikan performa yang lebih baik dengan R^2 tertinggi dan error terendah. Pengalaman kerja merupakan prediktor utama gaji, dengan hubungan non-linier yang jelas terlihat.

Rekomendasi:

- Gunakan model polinomial derajat 2 untuk prediksi gaji yang lebih akurat, tetapi perhatikan potensi *overfitting* pada data baru.
- Untuk pengembangan lebih lanjut, eksplorasi fitur tambahan seperti *Age* dan *Gender*, atau gunakan model lain seperti regresi ridge untuk mengatasi *overfitting*.

6 Daftar Pustaka

- Dataset "Salary Data" dari Kaggle: https://www.kaggle.com/datasets/mohithsairamreddy/salary-data