

## **50 Startups**

disusun untuk memenuhi  
tugas Pembelajaran Mesin

oleh :

Kelompok 4

Glenn Hakim	2208107010072
Ahmad Syah Ramadhan	2208107010033
Andika Pebriansyah	2208107010058
Nisa Rianti	2208107010018
Nuri Masyithah	2208107010006



**JURUSAN INFORMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS SYIAH KUALA**  
**2025**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam era digital yang semakin berkembang, startup memainkan peran penting dalam mendorong inovasi dan pertumbuhan ekonomi. Namun, kesuksesan sebuah startup tidak hanya ditentukan oleh ide yang inovatif, tetapi juga oleh faktor-faktor finansial dan strategis yang mendukung operasionalnya.

Dalam dunia bisnis, pemahaman tentang faktor-faktor yang mempengaruhi kesuksesan startup sangat penting. Dengan analisis data, kita dapat mengidentifikasi pola dan tren yang berkontribusi terhadap keberhasilan perusahaan rintisan. Dataset yang digunakan dalam analisis ini berasal dari Kaggle dan mencakup informasi tentang 50 startup, termasuk pengeluaran R&D, pemasaran, dan keuntungan yang diperoleh.

### **1.2 Tujuan**

Tujuan dari analisis ini adalah untuk:

1. Memahami faktor-faktor utama yang berkontribusi terhadap keberhasilan startup.
2. Menggunakan teknik analisis data untuk mengidentifikasi hubungan antara variabel dalam dataset.
3. Menerapkan model Machine Learning untuk memprediksi keuntungan startup berdasarkan variabel yang tersedia.

## BAB II

### PEMBAHASAN

#### 2.1 Pemahaman Dataset

Dataset yang digunakan dalam analisis ini adalah *50\_Startups.csv*, yang berisi informasi tentang investasi perusahaan rintisan dan keuntungan yang diperoleh. Variabel yang tersedia dalam dataset ini antara lain:

- R&D Spend: Investasi dalam penelitian dan pengembangan.
- Administration: Biaya administrasi perusahaan.
- Marketing Spend: Biaya pemasaran.
- State: Lokasi perusahaan (New York, California, Florida).
- Profit: Keuntungan yang diperoleh.

Statistik deskriptif dan visualisasi awal dilakukan untuk memahami distribusi data.

```
=====
STATISTIK DESKRIPTIF
=====
```

	R&D Spend	Administration	Marketing Spend	Profit
count	50.000000	50.000000	50.000000	50.000000
mean	73721.615600	121344.639600	211025.097800	112012.639200
std	45902.256482	28017.802755	122290.310726	40306.180338
min	0.000000	51283.140000	0.000000	14681.400000
25%	39936.370000	103730.875000	129300.132500	90138.902500
50%	73051.080000	122699.795000	212716.240000	107978.190000
75%	101602.800000	144842.180000	299469.085000	139765.977500
max	165349.200000	182645.560000	471784.100000	192261.830000

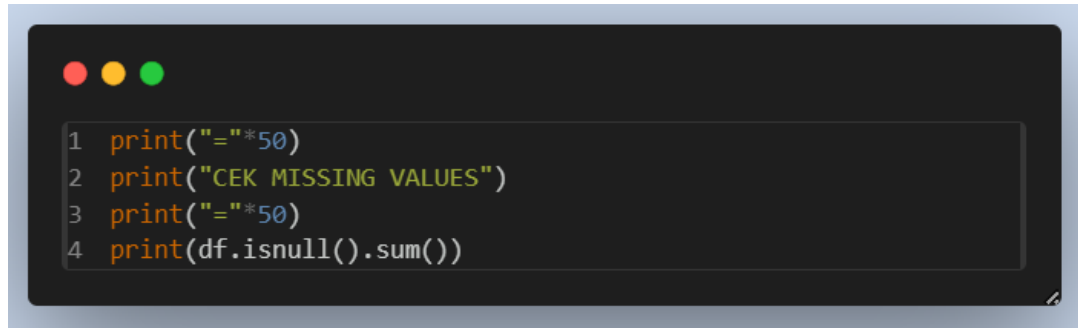
**Gambar 1.** Statistik deskriptif dan visualisasi awal dilakukan untuk memahami distribusi data.

Gambar di atas menunjukkan distribusi variabel dalam dataset. Kita dapat melihat bahwa investasi R&D dan pemasaran memiliki variasi yang lebih besar dibandingkan dengan biaya administrasi. Selain itu, distribusi profit cenderung mendekati distribusi normal.

## 2.2 Eksplorasi Data dan Pra-pemrosesan

Langkah-langkah yang dilakukan dalam eksplorasi data dan pra-pemrosesan meliputi:

- Mengecek *missing values*



```
1 print("="*50)
2 print("CEK MISSING VALUES")
3 print("="*50)
4 print(df.isnull().sum())
```

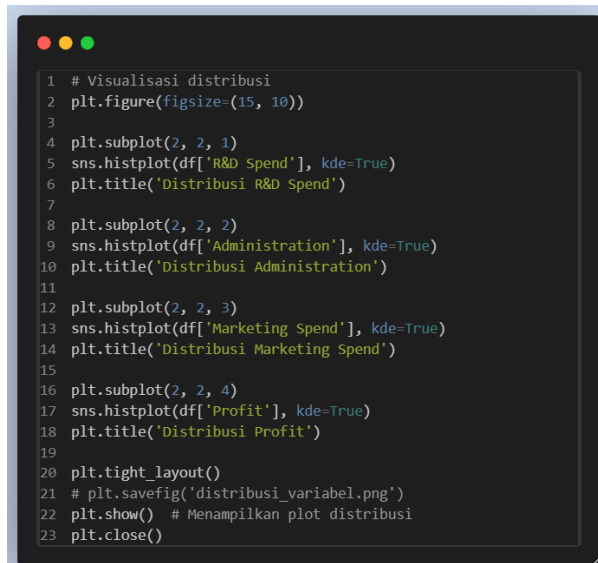
**Gambar 2.** Kode Mengecek Missing Values

```
=====
CEK MISSING VALUES
=====
R&D Spend      0
Administration 0
Marketing Spend 0
State           0
Profit          0
dtype: int64
```

**Gambar 3.** Output Mengecek Missing Values

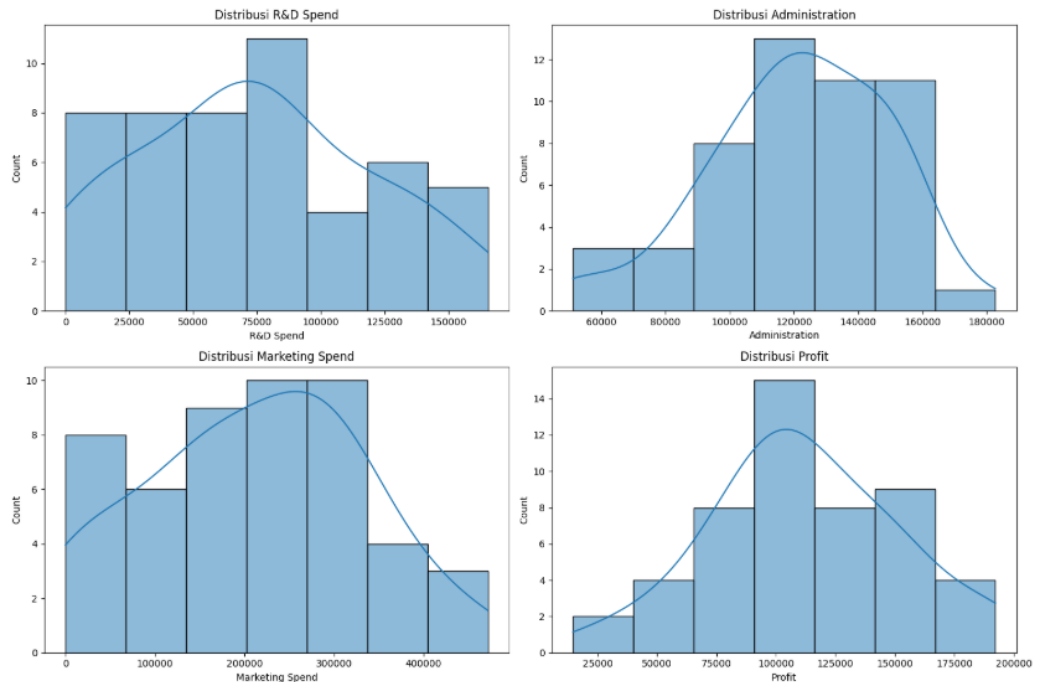
pada gambar diatas, tidak terdapat Missing Values pada dataset.

- Melihat distribusi pada data



```
1 # Visualisasi distribusi
2 plt.figure(figsize=(15, 10))
3
4 plt.subplot(2, 2, 1)
5 sns.histplot(df['R&D Spend'], kde=True)
6 plt.title('Distribusi R&D Spend')
7
8 plt.subplot(2, 2, 2)
9 sns.histplot(df['Administration'], kde=True)
10 plt.title('Distribusi Administration')
11
12 plt.subplot(2, 2, 3)
13 sns.histplot(df['Marketing Spend'], kde=True)
14 plt.title('Distribusi Marketing Spend')
15
16 plt.subplot(2, 2, 4)
17 sns.histplot(df['Profit'], kde=True)
18 plt.title('Distribusi Profit')
19
20 plt.tight_layout()
21 # plt.savefig('distribusi_variabel.png')
22 plt.show() # Menampilkan plot distribusi
23 plt.close()
```

**Gambar 4.** Kode distribusi pada data



**Gambar 5.** Visualisasi distribusi data dalam bentuk histogram

Pada gambar, menunjukkan distribusi empat variabel utama dalam dataset. **R&D Spend** dan **Marketing Spend** cenderung **positively skewed**, menunjukkan bahwa sebagian besar perusahaan mengalokasikan dana dalam jumlah sedang, tetapi ada beberapa yang menghabiskan jauh lebih besar. **Administration** memiliki distribusi yang lebih mendekati normal, dengan puncak di sekitar 120.000 - 140.000. **Profit** menunjukkan distribusi yang hampir simetris, mengindikasikan bahwa sebagian besar perusahaan memiliki profit di kisaran 100.000 - 125.000. Secara keseluruhan, pola distribusi ini dapat membantu memahami bagaimana pengeluaran di berbagai kategori berkorelasi dengan profitabilitas perusahaan.

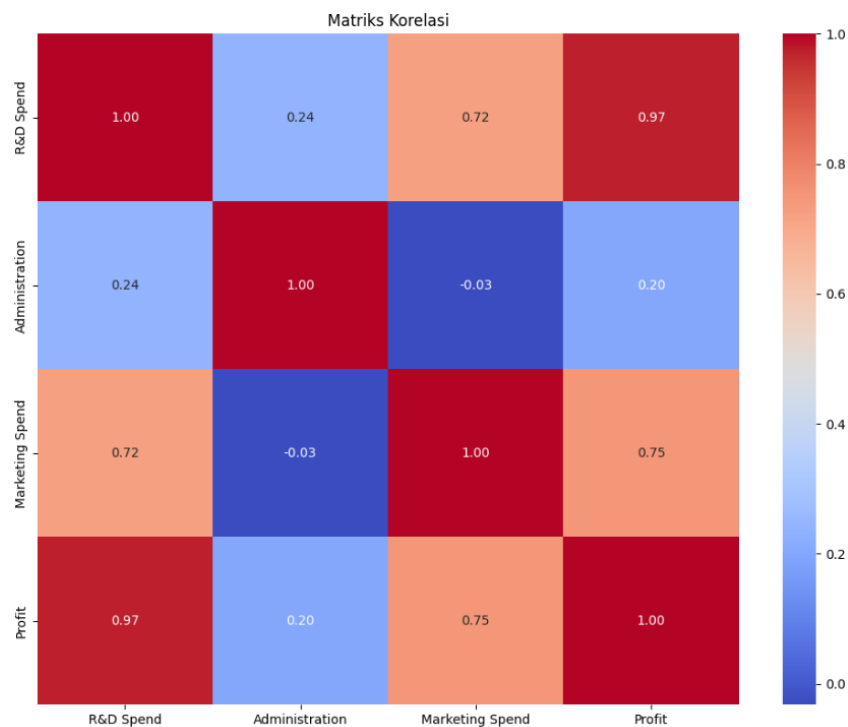
- Menganalisis hubungan antar variabel numerik menggunakan matriks korelasi

```

1 # Visualisasi hubungan antara variabel - HANYA UNTUK KOLOM NUMERIK
2 plt.figure(figsize=(10, 8))
3 # Gunakan hanya kolom numerik untuk korelasi
4 numeric_df = df.select_dtypes(include=['float64', 'int64'])
5 correlation = numeric_df.corr()
6
7 # Visualisasi dengan heatmap
8 sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt='.2f')
9 plt.title('Matriks Korelasi')
10 plt.tight_layout()
11 plt.show() # Menampilkan plot korelasi
12 plt.close()

```

**Gambar 6.** Kode Menganalisis hubungan antar variabel numerik menggunakan matriks korelasi



**Gambar 7.** Matriks Korelasi

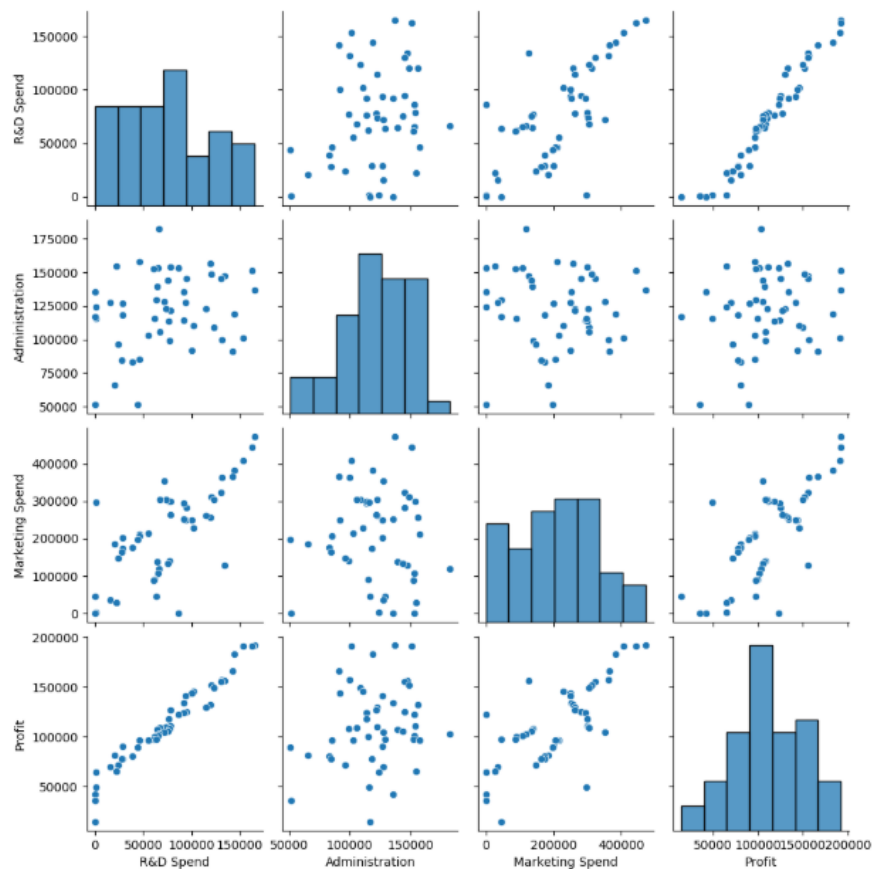
Pada gambar diatas, R&D Spend memiliki korelasi sangat kuat dengan Profit (0.97), menunjukkan bahwa investasi dalam penelitian berkontribusi besar

terhadap profit. Marketing Spend juga berpengaruh positif (0.75), meskipun tidak sekuat R&D Spend. Sementara itu, Administration memiliki korelasi rendah dengan Profit (0.20), menunjukkan dampak yang minimal. Hubungan antara Administration dan Marketing Spend (-0.03) hampir nol, menandakan tidak ada keterkaitan linier yang signifikan.

- **Mengecek Hubungan Distribusi data antar fitur dan label**

```
1 # Hubungan Distribusi data antar fitur dan label
2 sns.pairplot(df)
3 plt.show()
```

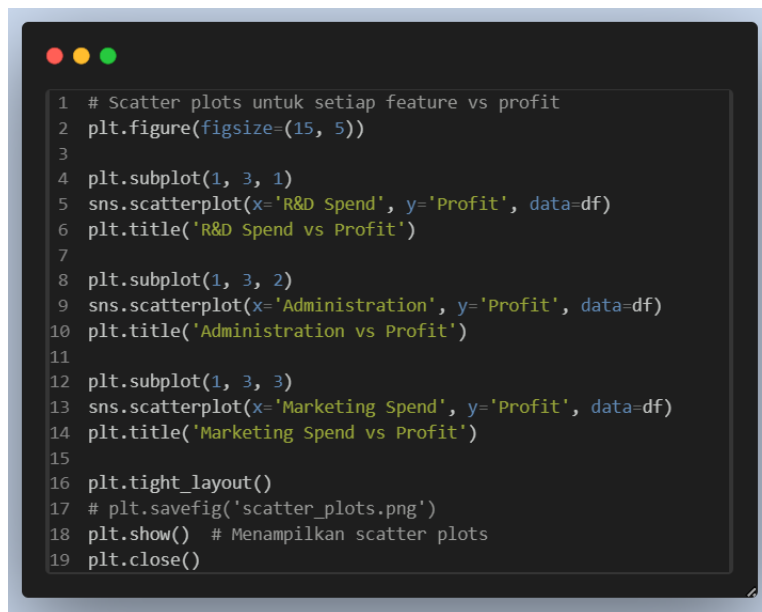
**Gambar 8.** Kode Hubungan Distribusi data antar fitur dan label



**Gambar 9.** Visualisasi Hubungan Distribusi data antar fitur dan label

Pada Plot pairplot diatas, menunjukkan hubungan antara variabel **R&D Spend**, **Administration**, **Marketing Spend**, dan **Profit**. Terlihat bahwa **R&D Spend** memiliki hubungan linear yang kuat dengan **Profit**, ditandai dengan pola titik yang membentuk garis diagonal ke atas. **Marketing Spend** juga menunjukkan korelasi positif dengan **Profit**, meskipun tidak sekuat R&D Spend. Sementara itu, **Administration** tidak menunjukkan hubungan yang jelas dengan variabel lainnya, terlihat dari sebaran titik yang acak. Distribusi masing-masing variabel juga ditampilkan dalam histogram di sepanjang diagonal, menunjukkan variasi data di setiap fitur.

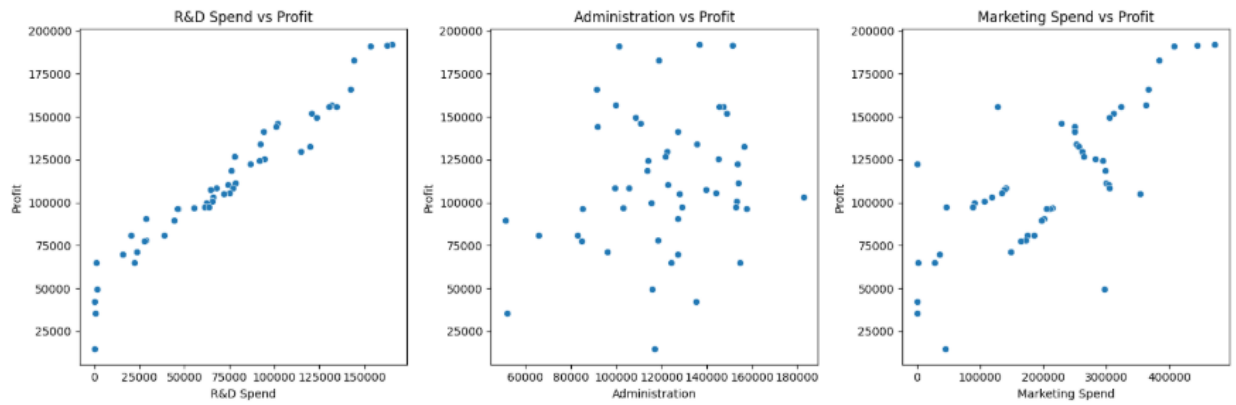
- Analisis Hubungan Pengeluaran dengan Profit dalam Bisnis



```
1 # Scatter plots untuk setiap feature vs profit
2 plt.figure(figsize=(15, 5))
3
4 plt.subplot(1, 3, 1)
5 sns.scatterplot(x='R&D Spend', y='Profit', data=df)
6 plt.title('R&D Spend vs Profit')
7
8 plt.subplot(1, 3, 2)
9 sns.scatterplot(x='Administration', y='Profit', data=df)
10 plt.title('Administration vs Profit')
11
12 plt.subplot(1, 3, 3)
13 sns.scatterplot(x='Marketing Spend', y='Profit', data=df)
14 plt.title('Marketing Spend vs Profit')
15
16 plt.tight_layout()
17 # plt.savefig('scatter_plots.png')
18 plt.show() # Menampilkan scatter plots
19 plt.close()
```

**Gambar 10.** Kode analisi hubungan pengeluaran dengan profit dalam bisnis





**Gambar 11.** Analisis Hubungan Pengeluaran Bisnis dengan

*Profit menggunakan scatter plot*

Pada gambar di atas, Scatter plot menunjukkan bahwa pengeluaran untuk R&D memiliki hubungan positif yang kuat dengan profit, ditunjukkan oleh pola titik yang membentuk tren linear. Sebaliknya, pengeluaran untuk administrasi tidak menunjukkan hubungan yang jelas dengan profit, karena titik-titiknya tersebar tanpa pola tertentu. Sementara itu, pengeluaran untuk pemasaran memiliki hubungan positif dengan profit, meskipun tidak sekuat hubungan antara R&D dan profit.

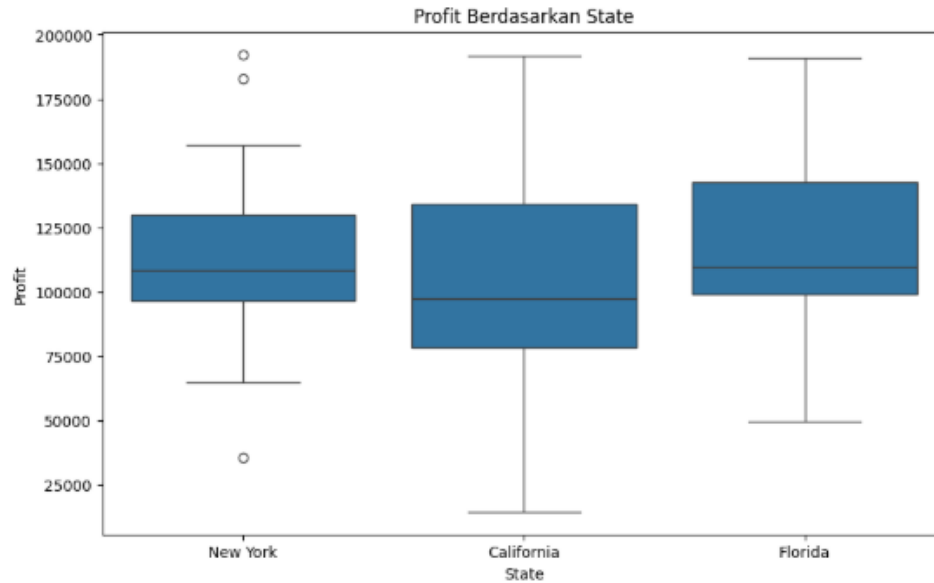
- **Melihat Distribusi profit berdasarkan state dengan Boxplot**

```

1 # Visualisasi profit berdasarkan state
2 plt.figure(figsize=(10, 6))
3 sns.boxplot(x='State', y='Profit', data=df)
4 plt.title('Profit Berdasarkan State')
5 # plt.savefig('profit_per_state.png')
6 plt.show() # Menampilkan boxplot
7 plt.close()

```

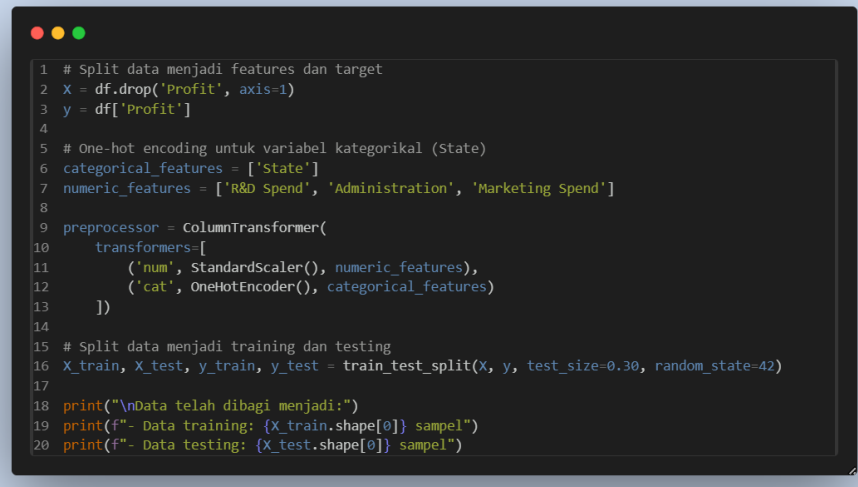
**Gambar 11.** Kode Distribusi Profit Berdasarkan State



**Gambar 12.** Visualisasi Distribusi Profit Berdasarkan State dengan Boxplot

Pada gambar di atas, Boxplot menunjukkan distribusi profit di tiga state: New York, California, dan Florida. California memiliki rentang profit yang lebih luas dibandingkan dua state lainnya, menunjukkan variabilitas profit yang lebih tinggi. Median profit di California lebih rendah dibandingkan Florida, tetapi memiliki pencilan yang lebih tinggi. New York memiliki distribusi yang lebih rapat dengan beberapa pencilan di bagian atas dan bawah, menunjukkan adanya beberapa perusahaan dengan profit yang sangat tinggi atau rendah dibandingkan mayoritas. Secara keseluruhan, Florida dan California memiliki profit yang lebih stabil dibandingkan New York.

- Pemrosesan data dan pembagian Dataset untuk pelatihan Model



```
1 # Split data menjadi features dan target
2 X = df.drop('Profit', axis=1)
3 y = df['Profit']
4
5 # One-hot encoding untuk variabel kategorikal (State)
6 categorical_features = ['State']
7 numeric_features = ['R&D Spend', 'Administration', 'Marketing Spend']
8
9 preprocessor = ColumnTransformer(
10     transformers=[
11         ('num', StandardScaler(), numeric_features),
12         ('cat', OneHotEncoder(), categorical_features)
13     ])
14
15 # Split data menjadi training dan testing
16 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42)
17
18 print("\nData telah dibagi menjadi:")
19 print(f"- Data training: {X_train.shape[0]} sampel")
20 print(f"- Data testing: {X_test.shape[0]} sampel")
```

**Gambar 13.** Kode pemrosesan data dan Pembagian dataset untuk pelatihan model

```
Data telah dibagi menjadi:
- Data training: 35 sampel
- Data testing: 15 sampel
```

**Gambar 14.** Output dari pembagian dataset untuk pelatihan model

Gambar menunjukkan pembagian dataset, di mana 35 sampel digunakan untuk training dan 15 sampel untuk testing, dengan rasio 70:30 menggunakan `train_test_split`.

## 2.3 Implementasi Model

```
1 param_grid_lr = {
2     'regressor__fit_intercept': [True, False],
3 }
4 linear_pipe = Pipeline([
5     ('preprocessor', preprocessor),
6     ('regressor', LinearRegression())
7 ])
8
9 grid_lr = GridSearchCV(linear_pipe, param_grid_lr, scoring='neg_mean_squared_error', cv=5)
10 grid_lr.fit(X_train, y_train)
11 y_pred_lr = grid_lr.best_estimator_.predict(X_test)
12
13 print("\nLinear Regression Metrics:")
14 print(f"Best Parameters: {grid_lr.best_params}")
15 print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred_lr):.4f}")
16 print(f"Root Mean Squared Error: {np.sqrt(mean_squared_error(y_test, y_pred_lr)):.4f}")
17 print(f"Mean Absolute Error: {mean_absolute_error(y_test, y_pred_lr):.4f}")
18 print(f"R-squared Score: {r2_score(y_test, y_pred_lr):.4f}")
19
20 y_pred_poly_degree = []
21
22 # GridSearch untuk Polynomial Regression dengan berbagai derajat
23 for degree in range(2, 6):
24     print(f"\nImplementasi Polynomial Regression (degree={degree}):")
25
26     poly_pipe = Pipeline([
27         ('preprocessor', preprocessor),
28         ('poly', PolynomialFeatures()),
29         ('regressor', LinearRegression())
30     ])
31
32     param_grid_poly = {
33         'poly__degree': [degree],
34         'poly__interaction_only': [True, False],
35         'regressor__fit_intercept': [True, False]
36     }
37     grid_poly = GridSearchCV(poly_pipe, param_grid_poly, scoring='neg_mean_squared_error', cv=5)
38     grid_poly.fit(X_train, y_train)
39
40     y_pred_poly = grid_poly.best_estimator_.predict(X_test)
41     y_pred_poly_degree.append(y_pred_poly)
42
43     print(f"Best Parameters: {grid_poly.best_params}")
44     print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred_poly):.4f}")
45     print(f"Root Mean Squared Error: {np.sqrt(mean_squared_error(y_test, y_pred_poly)):.4f}")
46     print(f"Mean Absolute Error: {mean_absolute_error(y_test, y_pred_poly):.4f}")
47     print(f"R-squared Score: {r2_score(y_test, y_pred_poly):.4f}")
48
```

**Gambar 15.** Kode implementasi model

Pada gambar di atas, kode melakukan pemodelan regresi linear dan regresi polinomial untuk memprediksi profit berdasarkan fitur yang telah diproses. Pertama, dilakukan pencarian hiperparameter terbaik untuk regresi linear menggunakan GridSearchCV, dengan parameter fit\_intercept. Model terbaik kemudian digunakan untuk membuat prediksi dan mengevaluasi performanya menggunakan metrik MSE, RMSE, MAE, dan R<sup>2</sup>. Selanjutnya, regresi polinomial diterapkan dengan derajat 2 hingga 5, menggunakan pipeline yang mencakup transformasi polynomial dan regresi linear. Hiperparameter terbaik untuk setiap derajat polinomial dicari menggunakan GridSearchCV, lalu model terbaik digunakan untuk prediksi dan evaluasi performa dengan metrik yang sama.

```

Linear Regression Metrics:
Best Parameters: {'regressor__fit_intercept': True}
Mean Squared Error: 84826955.0353
Root Mean Squared Error: 9210.1550
Mean Absolute Error: 7395.4335
R-squared Score: 0.9397

Implementasi Polynomial Regression (degree=2):
Best Parameters: {'poly__degree': 2, 'poly__interaction_only': True, 'regressor__fit_intercept': False}
Mean Squared Error: 97922059.2286
Root Mean Squared Error: 9895.5576
Mean Absolute Error: 9139.2124
R-squared Score: 0.9304

Implementasi Polynomial Regression (degree=3):
Best Parameters: {'poly__degree': 3, 'poly__interaction_only': False, 'regressor__fit_intercept': True}
Mean Squared Error: 25735619326.3971
Root Mean Squared Error: 160423.2506
Mean Absolute Error: 80920.0208
R-squared Score: -17.2911

Implementasi Polynomial Regression (degree=4):
Best Parameters: {'poly__degree': 4, 'poly__interaction_only': False, 'regressor__fit_intercept': True}
Mean Squared Error: 6337058448.2409
Root Mean Squared Error: 79605.6433
Mean Absolute Error: 41791.5915
R-squared Score: -3.5039

Implementasi Polynomial Regression (degree=5):
Best Parameters: {'poly__degree': 5, 'poly__interaction_only': False, 'regressor__fit_intercept': True}
Mean Squared Error: 6063615434.9339
Root Mean Squared Error: 77869.2201
Mean Absolute Error: 44847.2048
R-squared Score: -3.3096

```

**Gambar 16.** *Output dari Implementasi Model*

Pada gambar di atas, dapat dilihat berdasarkan hasil evaluasi, regresi linear memiliki performa yang cukup baik dengan nilai R-squared sebesar 0.9397, menunjukkan bahwa model mampu menjelaskan sekitar 93.97% variabilitas data. Pada regresi polinomial, model dengan derajat 2 memiliki performa terbaik dengan R-squared sebesar 0.9304, yang hampir setara dengan regresi linear. Namun, ketika derajat polinomial meningkat (derajat 3 ke atas), performa model justru menurun drastis, ditandai dengan meningkatnya error dan nilai R-squared yang negatif. Hal ini menunjukkan bahwa model mengalami overfitting, di mana model menjadi terlalu kompleks dan tidak mampu melakukan generalisasi dengan baik pada data uji.

## 2.4 Evaluasi Model

- Evaluasi Linear Regression



```
1 def evaluate_model(y_true, y_pred, model_name=None):
2     """
3     Comprehensive model evaluation function
4     """
5     mse = mean_squared_error(y_true, y_pred)
6     rmse = np.sqrt(mse)
7     mae = mean_absolute_error(y_true, y_pred)
8     mape = mean_absolute_percentage_error(y_true, y_pred)
9     r2 = r2_score(y_true, y_pred)
10
11     if model_name:
12         print(f"\nEvaluasi Model {model_name}:")
13         print(f"Mean Squared Error (MSE): {mse:.4f}")
14         print(f"Root Mean Squared Error (RMSE): {rmse:.4f}")
15         print(f"Mean Absolute Error (MAE): {mae:.4f}")
16         print(f"Mean Absolute Percentage Error (MAPE): {mape:.4f}%")
17         print(f"R2 Score: {r2:.4f}")
18
19     return {
20         'Model': model_name or 'Unnamed Model',
21         'MSE': mse,
22         'RMSE': rmse,
23         'MAE': mae,
24         'MAPE': mape,
25         'R2': r2
26     }
27
28 # Evaluasi Model
29 metrics = []
30
31 # 1. Evaluasi Linear Regression
32 print("\n=== EVALUASI LINEAR REGRESSION ===")
33 metrics.append(evaluate_model(y_test, y_pred_lr, "Linear Regression"))
```

**Gambar 17.** Kode evaluasi linear regression

```
=== EVALUASI LINEAR REGRESSION ===

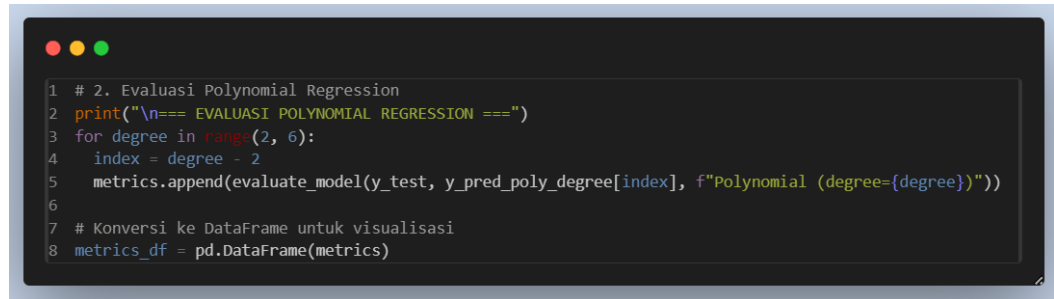
Evaluasi Model Linear Regression:
Mean Squared Error (MSE): 84826955.0353
Root Mean Squared Error (RMSE): 9210.1550
Mean Absolute Error (MAE): 7395.4335
Mean Absolute Percentage Error (MAPE): 0.0893%
R2 Score: 0.9397
```

**Gambar 18.** Output evaluasi linear regression

Pada gambar di atas, dapat dilihat evaluasi regresi linear menunjukkan model memiliki MSE sebesar 84.826.955,03 dan RMSE 9.210,16, menandakan tingkat kesalahan prediksi. MAE sebesar 7.395,43 menunjukkan rata-rata deviasi absolut, sementara MAPE yang sangat kecil (0,0893%) menunjukkan akurasi

tinggi. Dengan  $R^2$  sebesar 0,9397, model mampu menjelaskan 93,97% variabilitas data, menandakan performa prediksi yang sangat baik.

- **Evaluasi Polynomial Regression**



```
1 # 2. Evaluasi Polynomial Regression
2 print("\n=== EVALUASI POLYNOMIAL REGRESSION ===")
3 for degree in range(2, 6):
4     index = degree - 2
5     metrics.append(evaluate_model(y_test, y_pred_poly_degree[index], f"Polynomial (degree={degree})"))
6
7 # Konversi ke DataFrame untuk visualisasi
8 metrics_df = pd.DataFrame(metrics)
```

**Gambar 19.** *Evaluasi Polynomial Regression*

```
=== EVALUASI POLYNOMIAL REGRESSION ===

Evaluasi Model Polynomial (degree=2):
Mean Squared Error (MSE): 97922059.2286
Root Mean Squared Error (RMSE): 9895.5576
Mean Absolute Error (MAE): 9139.2124
Mean Absolute Percentage Error (MAPE): 0.0853%
R2 Score: 0.9304

Evaluasi Model Polynomial (degree=3):
Mean Squared Error (MSE): 25735619326.3971
Root Mean Squared Error (RMSE): 160423.2506
Mean Absolute Error (MAE): 80920.0208
Mean Absolute Percentage Error (MAPE): 1.3921%
R2 Score: -17.2911

Evaluasi Model Polynomial (degree=4):
Mean Squared Error (MSE): 6337058448.2409
Root Mean Squared Error (RMSE): 79605.6433
Mean Absolute Error (MAE): 41791.5915
Mean Absolute Percentage Error (MAPE): 0.5240%
R2 Score: -3.5039

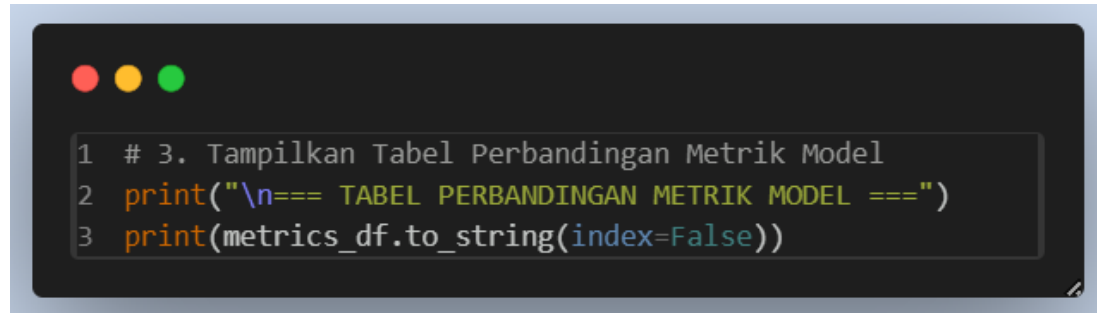
Evaluasi Model Polynomial (degree=5):
Mean Squared Error (MSE): 6063615434.9339
Root Mean Squared Error (RMSE): 77869.2201
Mean Absolute Error (MAE): 44847.2048
Mean Absolute Percentage Error (MAPE): 0.6806%
R2 Score: -3.3096
```

**Gambar 20.** *Evaluasi Polynomial Regression*

Pada gambar di atas, dapat dilihat hasil evaluasi Model polinomial derajat 2 memiliki performa terbaik ( $R^2 = 0,9304$ ). Derajat lebih tinggi ( $\geq 3$ ) menunjukkan overfitting dengan kesalahan prediksi meningkat dan  $R^2$  negatif.

## 2.5 Analisis Hasil

- **Tabel perbandingan Metrik Model**



```
1 # 3. Tampilkan Tabel Perbandingan Metrik Model
2 print("\n=== TABEL PERBANDINGAN METRIK MODEL ===")
3 print(metrics_df.to_string(index=False))
```

**Gambar 21.** Kode menampilkan tabel perbandingan metrik model

```
=== TABEL PERBANDINGAN METRIK MODEL ===
      Model      MSE      RMSE      MAE      MAPE      R2
Linear Regression 8.482696e+07  9210.154995  7395.433532  0.089299  0.939711
Polynomial (degree=2) 9.792206e+07  9895.557550  9139.212385  0.085270  0.930404
Polynomial (degree=3) 2.573562e+10 160423.250579  80920.020787  1.392137 -17.291117
Polynomial (degree=4) 6.337058e+09  79605.643319  41791.591546  0.524016 -3.503947
Polynomial (degree=5) 6.063615e+09  77869.220074  44847.204752  0.680648 -3.309603
```

**Gambar 22.** Output Tabel perbandingan metrik model

Regresi linear dan polinomial derajat 2 memiliki performa terbaik dengan  $R^2$  sekitar 0,93. Model polinomial dengan derajat  $\geq 3$  mengalami overfitting, ditunjukkan oleh  $R^2$  negatif dan peningkatan drastis pada MSE, RMSE, serta MAE.

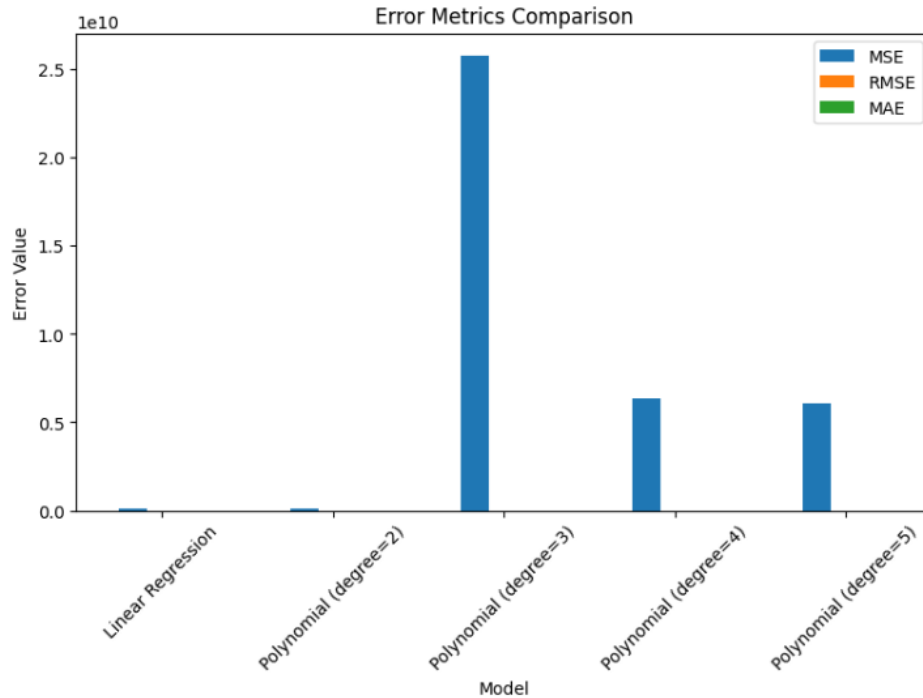
- **Visualisasi Perbandingan Metrik Error**



```
1 # 4. Visualisasi Perbandingan Metrik Error
2 plt.figure(figsize=(15, 10))
3 error_metrics = ['MSE', 'RMSE', 'MAE']
4
5 plt.subplot(2, 2, 1)
6 metrics_df.plot(x='Model', y=error_metrics, kind='bar', ax=plt.gca(), rot=45)
7 plt.title('Error Metrics Comparison')
8 plt.ylabel('Error Value')
9 plt.legend(loc='best')
10 plt.tight_layout()
```

**Gambar 23.** Kode visualisasi perbandingan Metrik Error





**Gambar 24.** Visualisasi perbandingan Metrik Error

Grafik menunjukkan bahwa regresi polinomial derajat  $\geq 3$  mengalami peningkatan drastis pada MSE, RMSE, dan MAE, mengindikasikan overfitting. Sementara itu, regresi linear dan polinomial derajat 2 memiliki error yang jauh lebih kecil, menunjukkan performa yang lebih stabil dan akurat.

- **Visualisasi  $R^2$  Score**

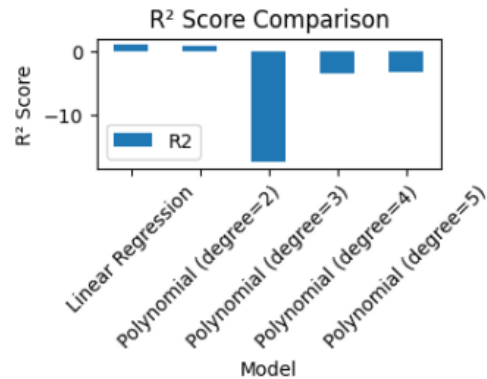
```

1 # 5. Visualisasi R2 Score
2 plt.subplot(2, 2, 2)
3 metrics_df.plot(x='Model', y=['R2'], kind='bar', ax=plt.gca(), rot=45)
4 plt.title('R2 Score Comparison')
5 plt.ylabel('R2 Score')
6 plt.tight_layout()
7
8 plt.suptitle('Model Performance Metrics', fontsize=16)
9 plt.tight_layout(rect=[0, 0.03, 1, 0.95])
10 plt.show()

```

**Gambar 25.** Kode visualisasi  $R^2$  Score

## Model Performance Metrics



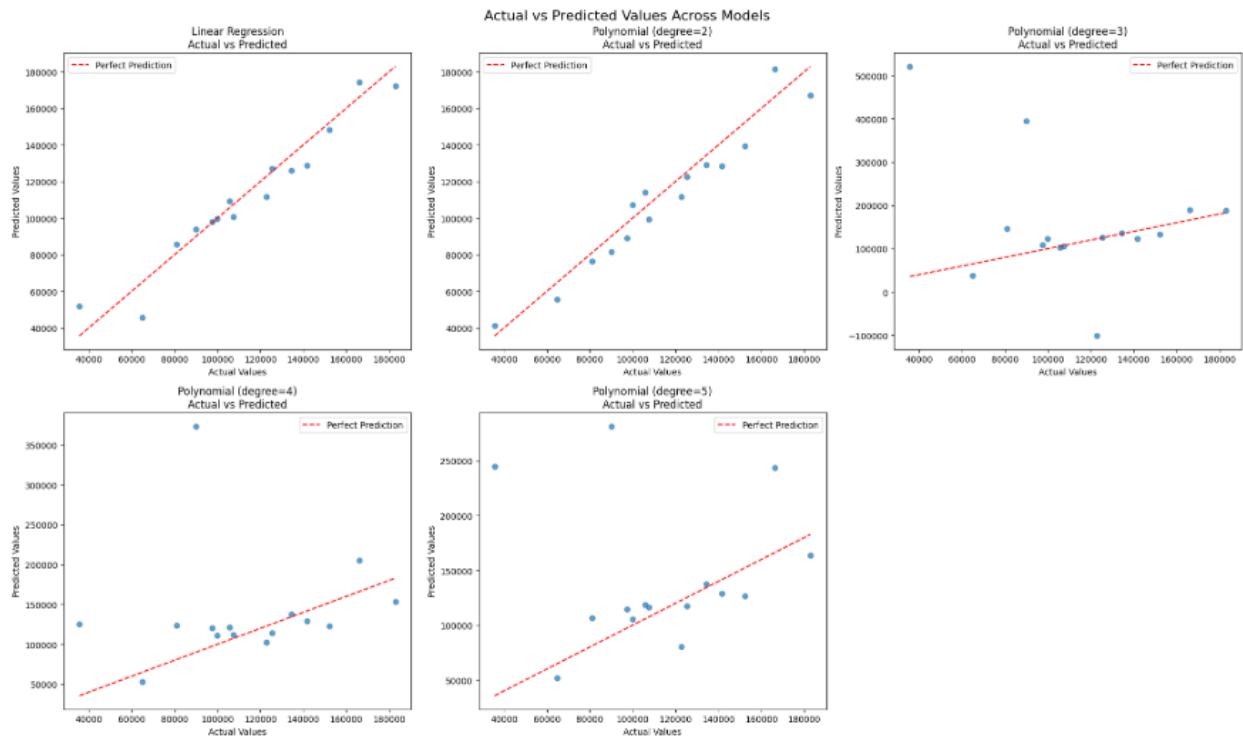
**Gambar 26.** Visualisasi  $R^2$  Score

Grafik menunjukkan bahwa regresi linear dan polinomial derajat 2 memiliki nilai  $R^2$  mendekati 1, menandakan performa yang baik. Sebaliknya, model polinomial dengan derajat  $\geq 3$  memiliki nilai  $R^2$  negatif, menunjukkan overfitting parah dan kegagalan dalam generalisasi data.

- Visualisasi Actual vs Predicted

```
1 # 6. Visualisasi Actual vs Predicted
2 plt.figure(figsize=(20, 12))
3
4 predictions = [y_pred_lr] + [y_pred_poly_degree[index_degree] for index_degree in range(0, 4)]
5 model_names = ['Linear Regression'] + [f'Polynomial (degree={degree})' for degree in range(2, 6)]
6
7 for i, (pred, name) in enumerate(zip(predictions, model_names), 1):
8     plt.subplot(2, 3, i)
9     plt.scatter(y_test, pred, alpha=0.7)
10    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', label='Perfect Prediction')
11    plt.title(f'{name}\nActual vs Predicted', fontsize=12)
12    plt.xlabel('Actual Values')
13    plt.ylabel('Predicted Values')
14    plt.legend()
15
16 plt.suptitle('Actual vs Predicted Values Across Models', fontsize=16)
17 plt.tight_layout()
18 plt.show()
```

**Gambar 27.** Kode visualisasi Actual vs Predicted



**Gambar 27.** Visualisasi Actual vs Predicted

Pada gambar di atas, Grafik menunjukkan bahwa regresi linear dan polinomial derajat 2 memiliki prediksi yang paling mendekati garis ideal (garis merah putus-putus). Sebaliknya, model polinomial dengan derajat  $\geq 3$  menunjukkan penyebaran yang semakin jauh dari garis ideal, menandakan prediksi yang buruk dan overfitting yang signifikan.

- **Identifikasi Model Terbaik**

```

1 # 7. Identifikasi Model Terbaik
2 best_model = metrics_df.loc[metrics_df['R2'].idxmax()]
3 print("\n=== MODEL TERBAIK ===")
4 print(f"Model Terbaik: {best_model['Model']}")
5 print("Alasan:")
6 print(f"- Memiliki R2 Tertinggi: {best_model['R2']:.4f}")
7 print(f"- MSE Terendah: {best_model['MSE']:.4f}")
8 print(f"- RMSE Terendah: {best_model['RMSE']:.4f}")

```

**Gambar 28.** Kode identifikasi Model Terbaik

```

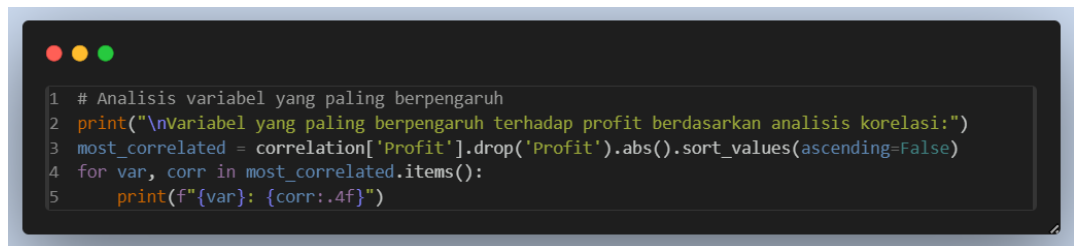
=== MODEL TERBAIK ===
Model Terbaik: Linear Regression
Alasan:
- Memiliki R2 Tertinggi: 0.9397
- MSE Terendah: 84826955.0353
- RMSE Terendah: 9210.1550

```

**Gambar 29.** *Output identifikasi Model Terbaik*

Regresi linear dipilih sebagai model terbaik karena memiliki R<sup>2</sup> tertinggi (0,9397), menunjukkan kecocokan terbaik dengan data. Selain itu, model ini memiliki nilai MSE dan RMSE terendah, menandakan kesalahan prediksi yang lebih kecil dibandingkan model lainnya.

- Analisis variabel yang paling berpengaruh



```

1 # Analisis variabel yang paling berpengaruh
2 print("\nVariabel yang paling berpengaruh terhadap profit berdasarkan analisis korelasi:")
3 most_correlated = correlation['Profit'].drop('Profit').abs().sort_values(ascending=False)
4 for var, corr in most_correlated.items():
5     print(f"{var}: {corr:.4f}")

```

**Gambar 30.** *kode analisis variabel yang paling berpengaruh*

```

Variabel yang paling berpengaruh terhadap profit berdasarkan analisis korelasi:
R&D Spend: 0.9729
Marketing Spend: 0.7478
Administration: 0.2007

```

**Gambar 31.** *Output Analisis variabel yang paling berpengaruh*

Berdasarkan analisis korelasi diatas, variabel yang paling berpengaruh terhadap profit adalah **R&D Spend** dengan korelasi sebesar **0.9729**, menunjukkan hubungan yang sangat kuat. **Marketing Spend** juga berpengaruh dengan korelasi **0.7478**, namun tidak sekuat R&D Spend. Sementara itu, **Administration** memiliki pengaruh paling lemah dengan korelasi **0.2007**, menandakan kontribusi yang kecil terhadap profit.

## 2.6 Kesimpulan

Kesimpulan dari analisis regresi ini menunjukkan bahwa **Linear Regression** adalah model terbaik dengan **R<sup>2</sup> Score sebesar 0.9397**, yang berarti model ini mampu menjelaskan **93.97% variasi dalam data**. Model regresi polinomial dengan derajat lebih tinggi ( $\text{degree} \geq 3$ ) justru memiliki nilai R<sup>2</sup> negatif, yang mengindikasikan bahwa model tersebut tidak cocok dan kemungkinan mengalami overfitting atau kesalahan dalam penyesuaian data.

Kesimpulan utama dari analisis ini adalah bahwa **model linear lebih efektif dibandingkan dengan model polinomial**, karena tetap memberikan hasil yang stabil tanpa overfitting.

Sebagai rekomendasi, disarankan untuk:

1. Memvalidasi model dengan data baru untuk memastikan performa yang konsisten.
2. Mempertimbangkan penggunaan teknik regularisasi jika ingin mengeksplorasi model yang lebih kompleks, guna menghindari overfitting.
3. Melakukan uji validasi silang untuk memastikan bahwa model bekerja dengan baik pada berbagai subset data.

Secara keseluruhan, pendekatan regresi linear menjadi pilihan yang lebih baik dalam menjelaskan hubungan antara variabel prediktor dan variabel target pada dataset ini.