

Perbandingan Algoritma Regresi Linear Sederhana dan Regresi Polinomial Dalam Prediksi Jumlah Penumpang Kereta Api Di Wilayah Jabodetabek

Ubaedillah Rohman^{1*}, Jefry Sunupurwa Asri², Hani Dewi Ariessanti³, Sawali Wahyu⁴

^{1,2,3,4} Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Esa Unggul Tangerang
Jl. Citra Raya Boulevard No.01 Blok S No.25, Panongan, Kec. Panongan, Kabupaten Tangerang, Banten 15711

^{1*}ubaedilah46@student.esaunggul.ac.id

²jefry.sunupurwa@esaunggul.ac.id

³hani.dewi@esaunggul.ac.id

⁴sawaliwahyu@esaunggul.ac.id

Intisari— Di era digital saat ini pertumbuhan data semakin mengalami peningkatan yang signifikan sehingga memerlukan solusi untuk analisis data yang efisien dan efektif. *Machine learning* menjadi solusi dikarenakan teknologi ini banyak digunakan di berbagai sektor. Diantara teknik *machine learning* algoritma regresi linear sederhana dan regresi polinomial berperan dalam melakukan analisis prediksi. Tujuan dari penelitian ini adalah membandingkan kinerja algoritma regresi linear sederhana dan regresi polinomial dalam prediksi jumlah penumpang kereta api di wilayah jabodetabek. Hasil penelitian akan menentukan manakah algoritma yang lebih akurat dalam memprediksi. Dari hasil penelitian regresi polinomial order 3 mendapatkan hasil akurasi tertinggi dengan R-Square 0.8983, MAPE 0.0704, dan RMSE 1446.6222 dibandingkan dengan regresi linear sederhana dengan R-Square 0.7943, MAPE 0.1013, dan RMSE 1713.8127.

Kata kunci— Perbandingan, Regresi linear sederhana, Regresi polinomial, Prediksi, Jumlah penumpang kereta

Abstract— In today's digital era, data growth is increasing significantly, requiring solutions for efficient and effective data analysis. Machine learning is the solution because this technology is widely used in various sectors. Among the machine learning techniques, simple linear regression algorithms and polynomial regression play a role in conducting predictive analysis. The purpose of this study is to compare the performance of simple linear regression algorithms and polynomial regression in predicting the number of train passengers in the Jabodetabek area. The results of the study will determine which algorithm is more accurate in predicting. From the results of the study, polynomial regression order 3 obtained the highest accuracy results with R-Square 0.8983, MAPE 0.0704, and RMSE 1446.6222 compared to simple linear regression with R-Square 0.7943, MAPE 0.1013, and RMSE 1713.8127.

Keywords— Comparison, Simple linear regression, Polynomial regression, Prediction, Number of train passengers.

I. PENDAHULUAN

Di era digital saat ini pertumbuhan data semakin mengalami peningkatan yang signifikan. Hal itu membuat analisis data menjadi semakin krusial di berbagai sektor sehingga memerlukan suatu solusi yang efisien dan efektif dalam menganalisis data. *Machine learning* menjadi solusi dikarenakan teknologi ini banyak digunakan di berbagai sektor. *Machine learning* menganalisis menggunakan algoritma statistik untuk melakukan berbagai analisis data seperti prediksi maupun klasifikasi. Diantara teknik-teknik machine learning, algoritma regresi linear sederhana dan regresi polinomial yang termasuk dalam kategori *supervised learning* berperan dalam melakukan analisis prediksi.

Regresi linear sederhana adalah persamaan hubungan antara satu variabel bebas dan variabel terikat yang diterangkan dengan garis linear [1]. Sedangkan regresi polinomial adalah

bentuk dari regresi linier berganda dengan menambahkan variabel prediktor yang dipangkatkan secara bertingkat hingga *order* ke- n [2]. Perbedaan dari kedua algoritma ini yaitu, regresi linear sederhana digunakan untuk menemukan pola data yang sederhana serta mudah diinterpretasikan. Sebaliknya, regresi polinomial dapat menangkap pola data yang lebih kompleks. Namun, dengan kekurangan semakin tinggi nilai *order* maka dapat mempengaruhi hasil prediksi.

Berdasarkan dari penjelasan diatas penelitian ini bertujuan untuk membandingkan kinerja algoritma regresi linear sederhana dan regresi polinomial dalam memprediksi jumlah penumpang kereta api di wilayah jabodetabek. Implementasi algoritma menggunakan Google Colab dan pemrograman Python. Fokus dari penelitian ini adalah mengevaluasi perbedaan pada proses serta hasil akurasi dari kedua algoritma tersebut. Hasil penelitian akan menentukan manakah algoritma terbaik untuk memprediksi jumlah penumpang kereta api serta

membantu memberikan referensi berguna untuk penelitian selanjutnya di sektor transportasi.

II. METODOLOGI PENELITIAN

A. Pengertian Machine learning

Menurut Christopher *Machine learning* adalah bidang ilmu yang mempelajari algoritma dan statistik yang memungkinkan sistem komputer menyelesaikan tugas tertentu tanpa instruksi langsung, dengan menggunakan pola dan proses inferensi. Algoritma *machine learning* membuat model matematika menggunakan data sampel yang disebut "*data training*", untuk menghasilkan prediksi tanpa diprogram secara langsung dalam melakukan tugas [3].

B. Python

Python adalah bahasa pemrograman yang disukai dikarenakan sederhana, kecil, dan fleksibel. Bahasa pemrograman ini banyak dipakai untuk misalnya, pengembangan perangkat lunak, pengembangan *web* dan *guide user interface* (GUI), serta pengembangan aplikasi desktop pendidikan dan ilmiah. Bahasa pemrograman ini tersedia di banyak platform seperti Windows, Linux, dan Mac OS [4].

C. Regresi linear sederhana

Regresi linear sederhana adalah persamaan hubungan antara satu variabel bebas atau prediktor (x) dan variabel terikat atau respon (y) dan diterangkan dengan garis linear [1]. Berikut di bawah ini persamaan regresi linear sederhana pada (1). Untuk mencari nilai a pada (2) dan b pada (3) [5]:

$$Y = \alpha + b.X \quad (1)$$

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2} \quad (2)$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \quad (3)$$

Keterangan :

α = konstanta

b = koefisien regresi

x = variabel *independen*/bebas

y = variabel *dependen*/terikat

D. Regresi polinomial

Regresi polinomial adalah bentuk dari regresi linear berganda dengan menambahkan variabel prediktor (x) yang dipangkatkan secara bertingkat hingga order ke-n [2]. Menurut James pada regresi polinomial nilai derajat yang digunakan biasanya kurang dari lima dikarenakan ketika nilai derajat menjadi naik model akan cenderung *fit* sehingga menimbulkan perilaku *overfitting* [6]. Berikut di bawah ini persamaan regresi polinomial pada (4) [7]:

$$Y = a_0 + b_1X + b_2X^2 + \dots + b_nX^n + \varepsilon \quad (4)$$

Apabila Y merupakan nilai yang diprediksi :

$$Y = a_0 + b_1X + b_2X^2 + \dots + b_nX^n \quad (5)$$

Dimana (y) sebagai variabel terikat, (a_0) sebagai intercept, (b_1, b_2, \dots, b_n) sebagai *slope*/koefisien regresi, (x) sebagai variabel bebas, (n) sebagai order/derajat, dan (ε) sebagai faktor error.

E. MAPE

Menurut Makridakis, Wheelwright, & Hyndman *Mean Absolute Percentage Error* adalah metrik yang sering dipakai untuk menilai kinerja peramalan. Metode ini menghitung rata-rata kesalahan absolut antara nilai sebenarnya dan prediksi dalam suatu periode. Metrik ini berguna untuk menilai akurasi maupun membandingkan pada model prediksi [8]. Bentuk persamaan MAPE dapat dilihat pada (6) [9]:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|Y_t - Y'_t|}{Y_t} \quad (6)$$

Keterangan :

Y_t = nilai *aktual* periode t

Y'_t = nilai prediksi periode t

n = banyaknya periode

Agar dapat mengetahui keakuratan suatu model dalam prediksi maka berikut pada tabel I interpretasi MAPE menurut Lewis dalam jurnal [10] :

TABEL I
INTERPRETASI MAPE

MAPE	Interpretation
< 10%	Prediksi yang sangat akurat
10-20%	Prediksi yang bagus
20-50%	Prediksi yang masuk akal
>50%	Prediksi yang tidak akurat

F. RMSE

Root Mean Square Error adalah metode untuk menghitung perbedaan antara nilai prediksi dan nilai sebenarnya. RMSE dihitung dengan mengambil akar kuadrat dari *Mean Square Error*. Nilai metode ini antara 0 sampai ∞ . Apabila model memiliki nilai RMSE yang rendah dapat diartikan lebih akurat dibandingkan model dengan nilai RMSE yang tinggi [11]. Bentuk persamaan RMSE dapat dilihat pada (7) [9] :

$$RMSE = \sqrt{\frac{1}{n} \sum (Y_t - Y'_t)^2} \quad (7)$$

Keterangan :

Y_t = nilai *aktual* periode t

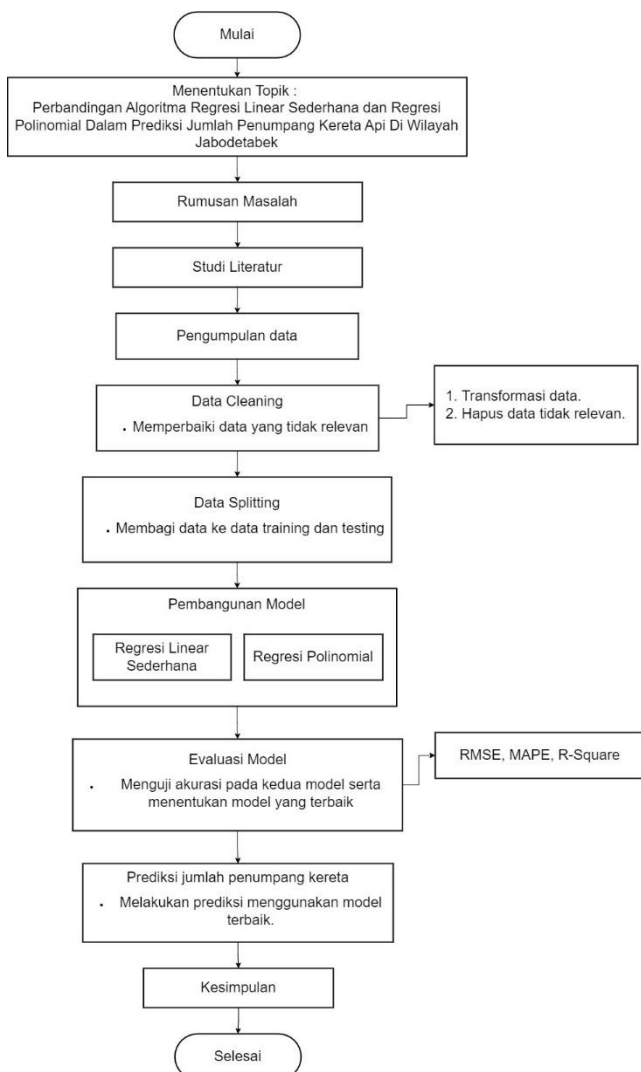
Y'_t = nilai prediksi periode t
 n = banyaknya periode

G. R-square

Koefisien determinasi atau disebut R Square maupun R kuadrat digunakan untuk mengukur besaran pengaruh variabel bebas (x) secara bersamaan terhadap variabel terikat (y). Ukuran nilai metode ini antara 0 hingga 1, dimana nilai yang mendekati satu menunjukkan bahwa model regresi memiliki kualitas yang baik. Bentuk persamaan R square dapat dilihat pada (8) [11] :

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - y_{avg})^2} \quad (8)$$

H. Alur penelitian



Gambar 1. Alur penelitian

Dalam proses membangun penelitian ini penulis membuat tahapan penelitian pada gambar 1. Berikut di bawah ini pemaparan dari proses penelitian yaitu,

1) Rumusan masalah

Berikut di bawah ini rumusan masalah penelitian.

1. Apakah algoritma regresi linear sederhana lebih efektif memprediksi jumlah penumpang kereta api dibandingkan dengan regresi polinomial?
2. Apa perbedaan hasil akurasi antara algoritma regresi linear sederhana dan regresi polinomial ketika melakukan pengujian?
3. Bagaimana proses implementasi algoritma regresi linear sederhana dan regresi polinomial di *machine learning*?

2) Studi literatur

Penulis melakukan studi literatur dengan mencari dan mengumpulkan beberapa literatur yang terkait dengan topik algoritma regresi linear sederhana maupun regresi polinomial untuk dianalisis dan digunakan sebagai pendukung penelitian.

3) Pengumpulan data

Data yang dikumpulkan adalah data penumpang kereta api di jabodetabek yang didapatkan dari situs badan pusat statistik. Proses pengumpulan data dilakukan dengan menyalin ke dalam file baru. Rentang data yang diambil tahun 2021-2022. Jumlah data terkumpul 24 *dataset*. Format pada data yaitu, numerik dan kategorik [12].

4) Data cleaning

Data yang dikumpulkan masih belum sempurna dan perlu diperbaiki agar dapat digunakan untuk pelatihan dan berikut ini aktivitas data *cleaning* yakni,

- Transformasi data.
- Hapus data tidak relevan.

5) Data splitting

Pada tahap ini fitur pada *dataset* dilabeli ke dalam variabel *independen* dan variabel *dependen*. Setelah itu *dataset* dibagi menjadi dua yaitu, data *training* dan data *testing*. Rasio data *training* sebesar 80% sedangkan data *testing* sebesar 20%.

6) Pembangunan model

Pada tahap ini algoritma regresi linear sederhana dan regresi polinomial akan dibangun menggunakan data *training*. Kedua algoritma ini memiliki perbedaan dari cara menggunakan variabel *independen* (x) untuk membangun model. Berikut pemaparan dari kedua algoritma.

1. Regresi linear sederhana

Pada algoritma regresi linear sederhana variabel *independen* (x) tidak memerlukan perubahan sehingga dapat melakukan pelatihan. Bentuk persamaan regresi linear sederhana dapat dilihat pada (1).

2. Regresi polinomial

Pada algoritma regresi polinomial variabel *independen* (x) diubah ke dalam nilai pangkat. Dalam hal ini regresi polinomial order yang digunakan adalah 2 dan 3. Setelah nilai diubah barulah dapat dilakukan pelatihan. Berikut ini persamaan regresi polinomial order 2 pada (9) dan order 3 pada (10).

$$Y = \alpha_0 + b_1 \cdot x^1 + b_2 \cdot x^2 + \varepsilon \quad (9)$$

$$Y = \alpha_0 + b_1 \cdot x^1 + b_2 \cdot x^2 + b_3 \cdot x^3 + \varepsilon \quad (10)$$

7) Evaluasi model

Kedua algoritma yang telah dibangun diuji keakuratannya menggunakan metode pengukur kesalahan yaitu, MAPE, RMSE, dan R-Square. Bagian ini memaparkan proses prediksi serta hasil akurasi yang akan menentukan manakah model terbaik.

8) Prediksi jumlah penumpang

Setelah ditentukan model yang terbaik maka dalam hal ini dilakukan prediksi jumlah penumpang kereta api di jabodetabek dalam 12 bulan (tahun 2023). Data yang dimiliki saat ini adalah 2021 dan 2022. Hasil prediksi ditampilkan dalam bentuk tabel.

III. HASIL DAN PEMBAHASAN

A. Pengumpulan data

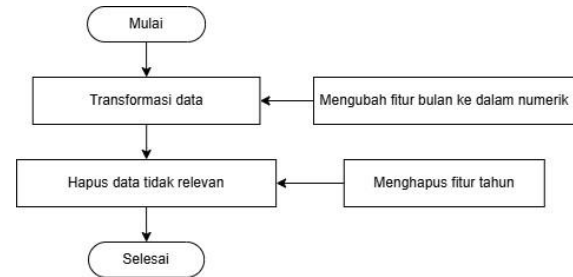
Data yang dikumpulkan adalah data penumpang kereta api di jabodetabek. Jumlah data sebanyak 24 dataset. Di dalam data berisi fitur tahun, bulan, dan jumlah penumpang. Proses pengumpulan data dilakukan dengan menyalin ke dalam file *datasheet*. Berikut gambar 2 hasil pengumpulan.

Tahun	Bulan	Jumlah_penumpang		
2021	Januari	10149		
2021	Februari	9796		
2021	Maret	12041		
2021	April	12452		
2021	Mei	12230		
2021	Juni	11978		
2021	Juli	5102		
2021	Agustus	5947		
2021	September	8693		
2021	Oktober	11347		
2021	November	12792		
2021	Desember	14213		
2022	Januari	14484		
2022	Februari	10499		
2022	Maret	15735		
2022	April	15890		
2022	Mei	17075		
2022	Juni	18326		
2022	Juli	19467		
2022	Agustus	19388		
2022	September	20587		
2022	Oktober	21807		
2022	November	21589		
2022	Desember	23118		

Gambar 2. Pengumpulan data

B. Data cleaning

Setelah data dikumpulkan selanjutnya dilakukan proses data *cleaning*. Proses data *cleaning* adalah melakukan perbaikan pada data yang belum sempurna sehingga dapat digunakan oleh model untuk pelatihan. Proses data *cleaning* disesuaikan dengan data yang dihadapi. Berikut keterangan di bawah ini.



Gambar 3. Alur data cleaning

1) Transformasi data

Pada gambar 2 pengumpulan data dapat dilihat bahwa fitur bulan memiliki nilai kategorik yang tidak bisa digunakan oleh model sehingga perlu diubah ke dalam nilai lain. Dalam hal ini nilai yang dipakai yaitu, *numerik*. Berikut gambar 4 hasil.

	Tahun	Bulan	Jumlah_penumpang
0	2021	1	10149
1	2021	2	9796
2	2021	3	12041
3	2021	4	12452
4	2021	5	12230
5	2021	6	11978
6	2021	7	5102
7	2021	8	5947
8	2021	9	8693
9	2021	10	11347
10	2021	11	12792
11	2021	12	14213

Gambar 4. Transformasi data

2) Hapus data tidak relevan

Pada gambar 4 dapat diketahui di dalam *dataset* terdapat fitur tahun, bulan dan jumlah penumpang. Dalam hal ini fitur yang akan digunakan yakni, bulan dan jumlah penumpang, sedangkan fitur tahun dihilangkan. Berikut gambar 5 hasil proses.

Bulan		Jumlah_penumpang
0	1	10149
1	2	9796
2	3	12041
3	4	12452
4	5	12230
5	6	11978
6	7	5102
7	8	5947
8	9	8693
9	10	11347
10	11	12792
11	12	14213

Gambar 5. Hapus fitur tidak relevan

C. Data splitting

Pada bagian ini setelah dilakukan data *cleaning* selanjutnya dilakukan proses data *splitting*. Sebelum dilakukan proses data *splitting* perlu ditentukan fitur yang akan menjadi variabel bebas dan terikat. Berikut keterangan pada tabel II.

TABEL II
FITUR VARIABEL

No	Fitur	Keterangan
1	Bulan	Variabel bebas (x1)
2	Jumlah penumpang	Variabel terikat (y)

Setelah menentukan variabel pada fitur, *dataset* kemudian dibagi ke dalam data *train* dan data *test*. Pada gambar 6 menampilkan jumlah data *train* dan *test* sebenarnya. Jumlah data *train* mendapatkan bagian sebanyak 19 *dataset*, sedangkan data *test* mendapatkan bagian sebanyak 5 *dataset*.

```
# Print data train dan data test
print('Jumlah data train x:',x_train.shape)
print('Jumlah data train y:',y_train.shape)
print('-----')
print('Jumlah data test x:',x_test.shape)
print('Jumlah data test y:',y_test.shape)
```

```
Jumlah data train x: (19, 1)
Jumlah data train y: (19,)
-----
Jumlah data test x: (5, 1)
Jumlah data test y: (5,)
```

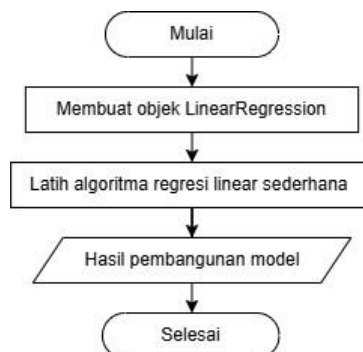
Gambar 6. Jumlah data train dan test

D. Pembangunan model

Pada bagian ini dilakukan pelatihan algoritma regresi linear sederhana dan regresi polinomial menggunakan data *train*. Berikut di bawah ini keterangan proses kedua algoritma.

1) Regresi linear sederhana

Untuk melakukan pelatihan maka perlu dipahami bagaimana prosesnya berjalan. Berikut gambar 7 alur pelatihan model dimulai dengan membuat objek *LinearRegression*. Kemudian latih algoritma menggunakan data *train*. Setelah itu maka model telah dibangun.



Gambar 7. Alur regresi linear

Pada pembangunan *machine learning* dilakukan dengan membuat objek *LinearRegression*. variabel *lin_reg* mewakili *LinearRegression* melakukan pelatihan menggunakan data *train*. Berikut gambar 8 *source code*.

```
# Latih model regresi linear sederhana
lin_reg = LinearRegression()
lin_reg.fit(x_train,y_train)
```

Gambar 8. Train regresi linear

Pada gambar 9 menampilkan nilai *intercept* dan *slope*, hasil dari pelatihan model. Diketahui nilai *intercept* (a) yakni, 7163,651369934365, sedangkan nilai *slope* yakni, 585,42363788. Perlu diketahui nilai ini digunakan sebagai parameter pada persamaan regresi linear sederhana sebagai berikut.

$$Y = 7163,651369934365 + 585,42363788.X$$

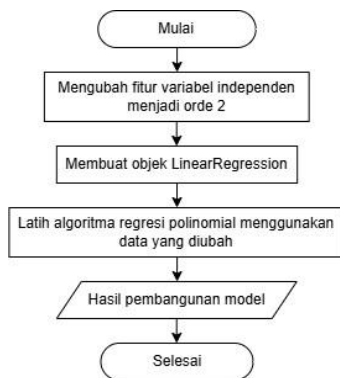
```
# Cetak nilai parameter intercept dan slope
print('Nilai intercept :',lin_reg.intercept_)
print('Nilai slope :',lin_reg.coef_)
```

```
Nilai intercept : 7163.651369934365
Nilai slope : [585.42363788]
```

Gambar 9. Parameter regresi linear

2) Regresi polinomial order 2

Berikut gambar 10 alur pelatihan model dimulai dengan mengubah fitur variabel independen di data *train* menjadi orde 2. Setelah itu, membuat objek *LinearRegression* dan latih regresi polinomial menggunakan data yang telah diubah. Setelah itu maka model telah dibangun.



Gambar 10. Alur polinomial orde 2

Pada pembangunan *machine learning* fitur variabel *independen* diubah menjadi orde 2 menggunakan objek *Polynomial Features*. Kemudian membuat objek *LinearRegression* untuk melatih menggunakan data yang telah diubah. Berikut gambar 11 *source code*.

```

# Modifikasi fitur polinomial orde 2
degree_2 = PolynomialFeatures(degree=2)
x2_train = degree_2.fit_transform(x_train)

# Latih model regresi polinomial
polinomial_2 = LinearRegression()
polinomial_2.fit(x2_train, y_train)
  
```

Gambar 11. Train polinomial orde 2

Pada gambar 12 menampilkan nilai *intercept* dan *slope*, hasil dari pelatihan model. Diketahui nilai *intercept* (a) yakni, 10752,467305008508, nilai *slope* (b1) = -276,12925339 dan (b2) = 35,03622489. Perlu diketahui nilai ini digunakan sebagai parameter pada persamaan regresi polinomial orde 2 sebagai berikut.

$$Y = 10752,467305008508 + -276,12925339.X + 35,03622489.X^2$$

```

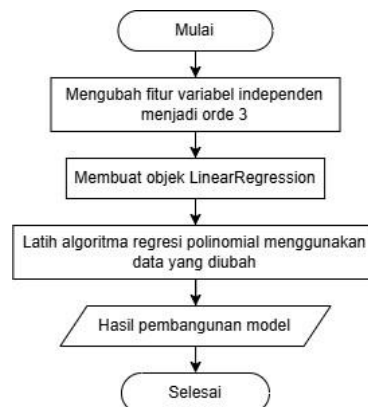
# Cetak nilai parameter intercept dan slope
print('Nilai intercept:', polinomial_2.intercept_)
print('Nilai slope:', polinomial_2.coef_)

Nilai intercept : 10752.467305008508
Nilai slope : [ 0.         -276.12925339  35.03622489]
  
```

Gambar 12. Parameter polinomial orde 2

3) Regresi polinomial order 3

Berikut gambar 13 alur pelatihan model dimulai dengan mengubah fitur variabel *independen* di data *train* menjadi orde 3. Setelah itu, membuat objek *LinearRegression* dan latih regresi polinomial menggunakan data yang telah diubah. Setelah itu maka model telah dibangun.



Gambar 13. Alur regresi polinomial orde 3

Pada pembangunan *machine learning* fitur variabel *independen* diubah menjadi orde 3 menggunakan objek *PolynomialFeatures*. Kemudian membuat objek *LinearRegression* untuk melatih menggunakan data yang telah diubah. Berikut gambar 14 *source code*.

```

# Modifikasi fitur polinomial orde 3
degree_3 = PolynomialFeatures(degree=3)
x3_train = degree_3.fit_transform(x_train)

# Latih model regresi polinomial
polinomial_3 = LinearRegression()
polinomial_3.fit(x3_train, y_train)
  
```

Gambar 14. Alur polinomial orde 3

Pada gambar 15 menampilkan nilai *intercept* dan *slope*, hasil dari pelatihan model. Diketahui nilai *intercept* (a) yakni, 12889,025531403375, sedangkan nilai *slope* (b1) = -1438,06942583 (b2) = 151,78545435 (b3) = -3,12691168. Perlu diketahui nilai ini digunakan sebagai parameter pada persamaan regresi polinomial orde 3 sebagai berikut.

$$Y = 12889,025531403375 + -1438,06942583.X + 151,78545435.X^2 + -3,12691168.X^3$$

```

# Cetak nilai parameter intercept dan slope
print('Nilai intercept:', polinomial_3.intercept_)
print('Nilai slope:', polinomial_3.coef_)

Nilai intercept : 12889.025531403375
Nilai slope : [ 0.         -1438.06942583  151.78545435 -3.12691168]
  
```

Gambar 15. Parameter polinomial orde 3

E. Evaluasi Model

Pada bagian ini dilakukan evaluasi pada kedua algoritma dengan menggunakan metode MAPE, RMSE, dan R-square. Berikut ini menerangkan proses prediksi serta hasil akurasi pada kedua algoritma. Hasil akurasi digunakan untuk menentukan manakah model yang berpengaruh.

1) Regresi linear sederhana

Pada gambar 16 proses prediksi dilakukan dengan menggunakan model yang sudah dilatih. variabel `lin_reg` mewakili model melakukan prediksi dengan input nilai `x_test`.

```
# Prediksi menggunakan data test
y_pred = lin_reg.predict(x_test)
```

Gambar 16. Prediksi regresi linear sederhana

Setelah mengetahui proses prediksi selanjutnya menampilkan hasil prediksi dalam bentuk tabel III serta implementasi perhitungan regresi linear sederhana menggunakan data *test* sebagai berikut.

Implementasi regresi linear sederhana :

$x = 9$

Solusi :

$$Y = \alpha + b \cdot X$$

$$y = 7163,651369934365 + 585,42363788 \cdot x$$

$$y = 7163,651369934365 + 585,42363788 \cdot (9)$$

$$y = 7163,651369934365 + 5.268,81274092$$

$$y = 12.432,464$$

Hasil prediksi bulan september (9) adalah 12.432

TABEL III
HASIL PREDIKSI REGRESI LINEAR

Input (x)	Data asli	Data prediksi (y)
17	17075	17116
18	18326	17701
20	19388	18872
15	15735	15945
9	8693	12432

2) Regresi polinomial order 2

Pada gambar 17 proses prediksi dilakukan dengan mengubah nilai `x_test` ke dalam orde 2. Setelah itu melakukan prediksi menggunakan data yang telah diubah untuk prediksi.

```
# Modifikasi fitur variabel independen
x2_test = degree_2.fit_transform(x_test)

# Prediksi menggunakan data modifikasi
y2_pred = polinomial_2.predict(x2_test)
```

Gambar 17. Prediksi polinomial orde 2

Setelah mengetahui proses prediksi selanjutnya menampilkan hasil prediksi dalam bentuk tabel IV serta implementasi perhitungan regresi polinomial orde 2 dilakukan menggunakan data *test* sebagai berikut.

Implementasi regresi polinomial orde 2 :

$x = 14$

$$x^2 = 14 \cdot 14 = 196$$

Solusi :

$$y = \alpha_0 + b_1 \cdot X^1 + b_2 \cdot X^2 + \varepsilon$$

$$y = 10752,467305008508 + -276,12925339 \cdot (14) + 35,03622489 \cdot (196)$$

$$y = 10752,467305008508 + -3.865,80954746 + 6.867,10007844$$

$$y = 6886,65775755 + 6.867,10007844$$

$$y = 13.753,757$$

Hasil prediksi bulan februari (14) adalah 13.753

TABEL IV
HASIL PREDIKSI POLINOMIAL ORDE 2

Input (x)	Data asli	Data prediksi (y)
14	10499	13754
16	15890	15304
2	9796	10340
21	20587	20405
11	12792	11954

3) Regresi polinomial order 3

Pada gambar 18 proses prediksi dilakukan dengan mengubah nilai `x_test` ke dalam orde 3. Setelah itu melakukan prediksi menggunakan data yang telah diubah untuk prediksi.

```
# Modifikasi fitur variabel independen
x3_test = degree_3.fit_transform(x_test)

# Prediksi menggunakan data modifikasi
y3_pred = polinomial_3.predict(x3_test)
```

Gambar 18. Prediksi regresi polinomial orde 3

Setelah mengetahui proses prediksi selanjutnya menampilkan hasil prediksi dalam bentuk tabel V serta implementasi perhitungan regresi polinomial orde 3 dilakukan menggunakan data *test* sebagai berikut.

Implementasi regresi polinomial orde 3 :

$x = 24$

$$x^2 = 24 \cdot 24 = 576$$

$$x^3 = 24 \cdot 24 \cdot 24 = 13.824$$

Solusi :

$$Y = \alpha_0 + b_1 \cdot X^1 + b_2 \cdot X^2 + b_3 \cdot X^3 + \varepsilon$$

$$y = 12889,025531403375 + -1438,06942583 \cdot (24) + 151,785 \cdot 45435 \cdot (576) + -3,12691168 \cdot (13.824)$$

$$y = 12889,025531403375 + -34.513,66621992 + 87.428,4217056 + -43.226,42706432$$

$$y = -21624,6406885 + 87.428,4217056 + -43.226,42706432$$

$$y = -21624,6406885 + 44201,9946413$$

$$y = 22.577,353$$

Hasil prediksi bulan ke desember (24) adalah 22.577

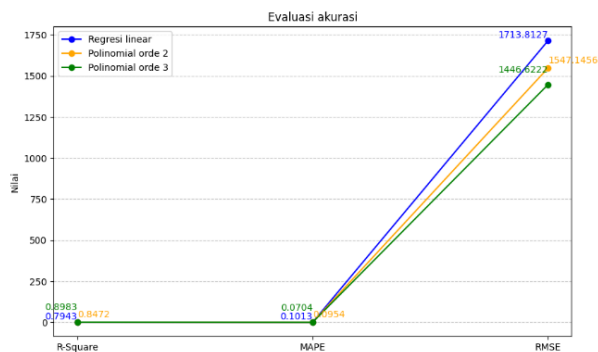
TABEL V
HASIL PREDIKSI POLINOMIAL ORDE 3

Input (x)	Data asli	Data prediksi (y)
24	23118	22577
4	12452	9365
21	20587	20669
17	17075	16945

10	11347	10560
----	-------	-------

Desember	36	11943
----------	----	-------

Setelah melihat proses perhitungan selanjutnya di bawah ini dilakukan analisa akurasi pada kedua algoritma. Untuk dapat lebih jelas berikut di bawah ini menampilkan bentuk grafik pada gambar 19 dan tabel VI. Dari keseluruhan akurasi pada MAPE, RMSE, dan R-square dapat disimpulkan regresi polinomial orde 3 sebagai model yang berpengaruh dibandingkan dengan algoritma lainnya.



Gambar 19. Grafik evaluasi model

TABEL VI
HASIL EVALUASI AKURASI

	Linear	Order 2	Order 3
R-square	0.7943	0.8471	0.8983
MAPE	0.1013	0.0953	0.0704
RMSE	1713.8127	1547.1456	1446.6222

F. Prediksi jumlah penumpang

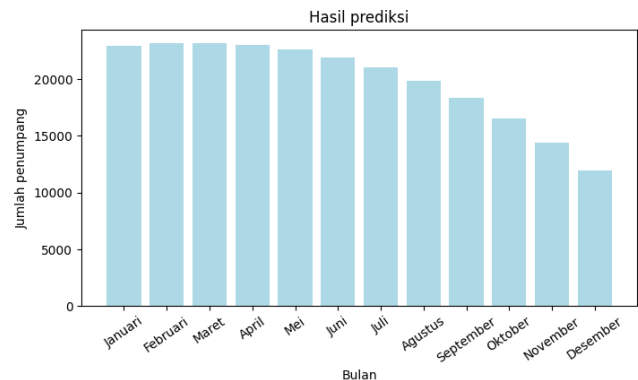
Pada bagian ini dilakukan prediksi jumlah penumpang kereta api menggunakan regresi polinomial order 3. Data yang dimiliki saat ini adalah 2021 dan 2022. Prediksi yang akan diprediksi yaitu, tahun 2023 dari bulan januari hingga februari. Data yang di input untuk prediksi yaitu 25-36. Berikut di bawah ini hasil prediksi pada tabel VII.

TABEL VII
PREDIKSI JUMLAH PENUMPANG

Bulan	Input	Jumlah penumpang
Januari	25	22945
Februari	26	23148
Maret	27	23166
April	28	22981
Mei	29	22574
Juni	30	21927
Juli	31	21021
Agustus	32	19836
September	33	18355
Oktober	34	16559
November	35	14427

G. Grafik jumlah penumpang

Pada gambar 20 diketahui prediksi jumlah penumpang kereta api terjadi penurunan pada bulan mei hingga bulan desember. Hal ini memberikan perkiraan bahwa di bulan tersebut akan mengalami penurunan jumlah penumpang secara drastis.



Gambar 20. Grafik jumlah penumpang

IV. KESIMPULAN

Dari hasil penelitian diatas penulis dapat menyimpulkan beberapa poin :

1. Pada pembangunan dan prediksi, algoritma regresi linear sederhana tidak mengubah data *train* dan data *test* ke dalam bentuk lain, sebaliknya algoritma regresi polinomial mengubah data pada variabel *independen* menjadi nilai derajat ke-n.
2. Berdasarkan hasil evaluasi pada kedua algoritma dapat disimpulkan bahwa regresi polinomial orde 3 yang paling berpengaruh dibandingkan dengan regresi lainnya seperti polinomial orde 2 dan regresi linear sederhana sehingga dapat digunakan untuk prediksi jumlah penumpang kereta api di jabodetabek.

REFERENSI

- [1] J. Raharjo, B. Rahmat, and J. Hasudungan, "Monitoring Indonesia's Energy Mix Achievement Using Simple Linear Regression," *International Journal of Energy Economics and Policy*, vol. 12, no. 2, 2022, doi: 10.32479/ijeep.12759.
- [2] N. Y. I. Ginting, A. Novianty, and A. L. Prasasti, "ESTIMASI MAGNITUDO GEMPA BUMI DARI SINYAL SEISMIC GELOMBANG P MENGGUNAKAN METODE REGRESI POLINOMIAL," *e-Proceeding of Engineering*, vol. 7, no. 2, pp. 4633–4642, Aug. 2020.
- [3] R. R. Pratama, "Analisis Model Machine Learning Terhadap Pengenalan Aktifitas Manusia," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 19, no. 2, pp. 302–311, May 2020, doi: 10.30812/matrik.v19i2.688.
- [4] S. Anam, F. Widhiatmoko, I. Yanti, Z. Fitriah, U. Sa'adah, and A. N. Guci, *Pengantar Algoritma dan Pemrograman dengan Python*. 2023. doi: 10.11594/ubpress9786232967342.

- [5] Harsiti, Z. Muttaqin, and E. Srihartini, "PENERAPAN METODE REGRESI LINIER SEDERHANA UNTUK PREDIKSI PERSEDIAAN OBAT JENIS TABLET," *JSiI (Jurnal Sistem Informasi)*, vol. 9, no. 1, pp. 12–16, Mar. 2022, doi: 10.30656/jsii.v9i1.4426.
- [6] J. Pardede, "MIND (Multimedia Artificial Intelligent Networking Database House Prices Prediction: Multiple Linear Regression vs Ridge vs Polynomial," *Journal MIND Journal / ISSN*, vol. 8, no. 1, 2023.
- [7] A. Eka Putra and A. Juarna, "Prediksi Pro duksi Daging Sapi Nasional dengan Meto de Regresi Linier dan Regresi Polinomial," *Jurnal Ilmiah Komputasi*, vol. 20, no. 2, 2021, doi: 10.32409/jikstik.20.2.2722.
- [8] E. Giovany Syuhada and M. Y. Helmi Setyawan, "ANALISIS KOMPARASI METODE PROPHET DAN METODE EXPONENTIAL SMOOTHING DALAM PERAMALAN JUMLAH PENGANGGURAN DI JAWA BARAT: SYSTEMATIC LITERATURE REVIEW," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 2, 2023, doi: 10.36040/jati.v7i2.6827.
- [9] A. Ghebyla Najla and D. Fitrianah, "Penerapan Metode Regresi Linear Untuk Prediksi Penjualan Properti pada PT XYZ," *Jurnal Telematika*, vol. 14, no. 2, pp. 79–86, 2019.
- [10] J. J. Montaña Moreno, A. Palmer Pol, A. Sesé Abad, and B. Cajal Blasco, "Using the R-MAPE index as a resistant measure of forecast accuracy.," *Psicothema*, vol. 25, no. 4, pp. 500–6, 2013, doi: 10.7334/psicothema2013.23.
- [11] I. Daqiqil, *Machine Learning : Teori, Studi Kasus dan Implementasi Menggunakan Python*. Riau: UR PRESS, 2021. doi: 10.5281/zenodo.5113507.
- [12] "Badan Pusat Statistik Indonesia," Jumlah Penumpang Kereta Api. Accessed: Dec. 05, 2023. [Online]. Available: <https://www.bps.go.id/id/statistics-table/2/NzIjMg%3D%3D/jumlah-penumpang-kereta-api.html>