
Perbandingan Metode *Simple Linear Regression*, *Polynomial Regression*, dan *K-Nearest Neighbors (KNN)* untuk Prediksi Transaksi Perbankan

Muhammad Bagas Fadilah¹, Luthfansa Alfaqih², dan Ardiansyah Dores^{3*}

^{1,2,3}Fakultas Ilmu Komputer, Program Studi Sistem Informasi,
Universitas Mercubuana

*ardian@.mercubuana.ac.id

Abstrak

Dalam era digital, sektor perbankan menghadapi tantangan kompleks dalam mengelola data transaksi dan pengambilan keputusan strategis. Bank XYZ, seperti banyak bank lain, mengumpulkan banyak data transaksi harian yang mencakup volume, waktu, jenis transaksi, dan profil pelanggan. Meskipun data ini memiliki potensi besar untuk memberikan wawasan berharga, seringkali belum dimanfaatkan optimal untuk mendukung pengambilan keputusan yang lebih cerdas dan responsif. Penelitian ini membahas keefektifan tiga metode prediksi dalam analisis transaksi perbankan, yaitu *Simple Linear Regression*, *Polynomial Regression*, dan *K-Nearest Neighbors (KNN)*. Data transaksi dikumpulkan dan diproses melalui data *cleaning*. Setelah itu, model prediksi dibangun dan diuji menggunakan data *training* dan *testing* data. Hasil prediksi dari ketiga model dibandingkan menggunakan *Root Mean Square Error (RMSE)* untuk menentukan model dengan performa terbaik. Berdasarkan hasil pengujian ketiga metode menunjukkan bahwa metode *Polynomial Regression* memiliki hasil yang lebih baik untuk nilai *RMSE* 0,1724, sedangkan untuk metode *Simple Linear Regression* mendapatkan nilai *RMSE* 0,1844, dan metode *K-Nearest Neighbors (KNN)* mendapatkan nilai *RMSE* 0,1739. Kesimpulan nya adalah berdasarkan nilai *RMSE* 3 metode diatas, *Polynomial Regression* lebih unggul dibandingkan metode *Simple Linear Regression* dan *K-Nearest Neighbors (KNN)*.

Kata Kunci: *Simple Linear Regression*, *Polynomial Regression*, *K-Nearest Neighbors (KNN)*, Prediksi, Data *Cleaning*, *RMSE*

Abstract

In the digital era, the banking sector faces complex challenges in managing transaction data and making strategic decisions. Bank XYZ, like many other banks, collects a large amount of daily transaction data that includes volume, time, transaction type, and customer profiles. Although this data has great potential to provide valuable insights, it is often not fully utilized to support smarter and more responsive decision-making. This study examines the effectiveness of three prediction methods in banking transaction analysis: Simple Linear Regression, Polynomial Regression, and K-Nearest Neighbors (KNN). Transaction data was collected and processed through data cleaning. Subsequently, predictive models were built and tested using training data and testing data. The prediction results from the three models were compared using Root Mean Square Error (RMSE) to determine the best-performing model. Based on the testing results, Polynomial Regression showed the best performance with an RMSE of 0.1724, while Simple Linear Regression had an RMSE of 0.1844, and K-Nearest Neighbors (KNN) had an RMSE of 0.1739. The conclusion is that, based on the RMSE values of the three methods, Polynomial Regression outperformed both Simple Linear Regression and K-Nearest Neighbors (KNN).

Keywords: *Simple Linear Regression, Polynomial Regression, K-Nearest Neighbors (KNN), Prediction, Data Cleaning, RMSE*

1. Pendahuluan

Dalam era transformasi digital, perbankan telah mengalami perubahan signifikan dengan adopsi teknologi canggih untuk meningkatkan efisiensi dan akurasi operasional. Salah satu aspek penting dalam perbankan modern adalah kemampuan untuk memprediksi transaksi keuangan dengan akurat. Prediksi transaksi perbankan tidak hanya membantu dalam manajemen risiko, tetapi juga memungkinkan bank untuk mengoptimalkan sumber daya dan meningkatkan layanan pelanggan. Beberapa metode yang umum digunakan untuk prediksi transaksi meliputi *Simple Linear Regression* (Hasanudin, M., Prihandi, I., 2024), (Safitri dan Sari, 2023), (Ariesanto, 2020), (Sholeh et al., 2023), (Wibow et al., 2023), (Husdi dan Dalai, 2023), *Polynomial Regression* (Satriyo et al., 2023), (Fauzi et al., 2023), (Rachma et al., 2024), dan *K-Nearest Neighbors (KNN)* (Priambodo, B., 2021), (Priambodo, B., 2020), (Priambodo, B., 2019), (Pratiwi et al., 2023), (Nurfauzan, 2022), (Alfat, 2022). *Simple Linear Regression* merupakan metode statistik yang digunakan untuk menghubungkan hubungan linear antara variabel independen dan variabel dependen (Montgomery, Peck, & Vining, 2021). Metode ini sederhana dan mudah diterapkan, namun mungkin tidak selalu memberikan hasil yang akurat ketika hubungan antar variabel tidak linear. *Polynomial Regression* adalah bentuk pengembangan dari *Simple Linear Regression* yang memungkinkan hubungan non-linear antara variabel independen dan dependen dengan memasukkan derajat polinomial yang lebih tinggi (Draper & Smith, 1998). Meskipun lebih fleksibel, *Polynomial Regression* dapat menjadi *overfitting* jika derajat polinomial yang dipilih terlalu tinggi.

Sedangkan, *K-Nearest Neighbors (KNN)* adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi (Cover & Hart, 1967). Metode ini bekerja dengan mencari sejumlah k tetangga terdekat untuk membuat prediksi berdasarkan mayoritas atau rata-rata dari tetangga-tetangga tersebut. *KNN* sangat efektif dalam menangani data non-linear, namun kinerja algoritma ini bisa menurun seiring dengan meningkatnya jumlah

dimensi data.

Penelitian ini bertujuan untuk membandingkan kinerja dari ketiga metode prediksi tersebut dalam konteks transaksi perbankan. Dengan menganalisis dan membandingkan keakuratan, kecepatan, dan kompleksitas dari *Simple Linear Regression*, *Polynomial Regression*, dan *K-Nearest Neighbors*, diharapkan dapat memberikan wawasan mengenai metode mana yang paling efektif digunakan dalam prediksi transaksi perbankan.

RMSE adalah metode yang digunakan untuk mengukur perbedaan antara nilai yang diprediksi oleh sebuah model dan nilai yang diobservasi. Keakuratan estimasi dinilai berdasarkan nilai *RMSE* yang kecil. Semakin kecil nilai *RMSE* pada suatu estimasi, semakin akurat estimasi tersebut dibandingkan dengan nilai *RMSE* yang lebih besar. Dalam penelitian yang menggunakan *machine learning* untuk prediksi data, parameter *RMSE* sangat diperlukan untuk menilai akurasi suatu model. (Nurfauzan, 2022), (Satriyo et al., 2023), Mulyana, D. I. (2022).

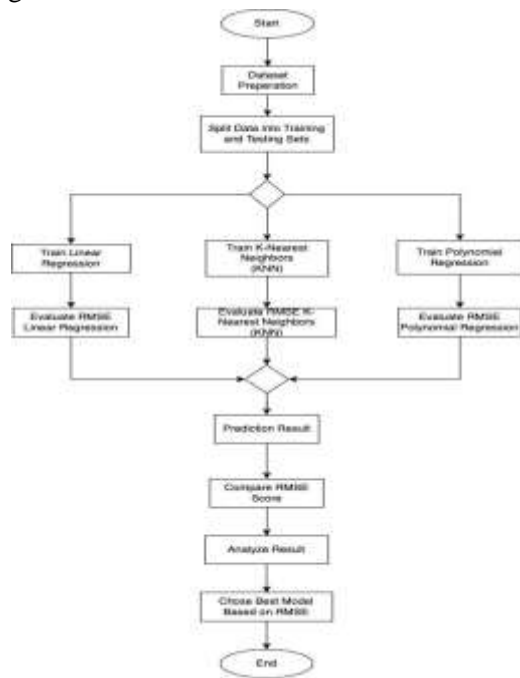
Seiring berkembangnya teknologi, penerapan metode *data mining* dapat dilakukan untuk melakukan prediksi cuaca sehingga mendapatkan nilai akurasi yang baik. *Data mining* merupakan rangkaian proses pengumpulan data dalam skala besar. Inti dari *data mining* untuk mempermudah proses pengambilan keputusan dengan prediksi yang akurat. Manfaat dari metode ini terletak pada kemudahannya dalam menganalisis dataset yang luas (Arif & Nurmaleni, 2023; Jayadianti et al., 2020; Nurul Salsabila Syam et al., 2022). Salah satu metode yang digunakan dalam data mining adalah *machine learning*.



Gambar 1. Grafik transaksi pada bank XYZ di bulan Januari 2024 – November 2024 pukul 00.00

2. Metode Penelitian

Penelitian ini menggunakan data kuantitatif yang bersifat numerik. Data yang dianalisis dalam penelitian ini merupakan data transaksi perbankan XYZ yang dikumpulkan dari bulan Januari 2024 hingga November 2024. Berikut diagram alir penelitian pada gambar 2.



Gambar 2. Diagram Alir Penelitian

Pengumpulan Data

Langkah pertama adalah mengumpulkan data set yaitu data transaksi perbankan. Pengumpulan data dilakukan pada tanggal 30 November 2024, dengan data berasal dari Bank XYZ. Data yang dikumpulkan meliputi transaksi perbankan dari 01 Januari 2024 hingga 30 November 2024 pada pukul 00:00. Data yang telah dikumpulkan akan dianalisis untuk menghasilkan informasi baru yang dapat digunakan dalam memprediksi masalah yang diteliti. Data ini akan dijadikan basis untuk membangun model prediksi.

Preprocessing Data

Preprocessing data adalah tahap penting dalam memastikan kualitas data yang digunakan untuk model prediksi. Proses ini mencakup langkah-langkah untuk melakukan data *cleaning*. Data *cleaning* adalah proses pembersihan data untuk memastikan bahwa data bebas dari kesalahan dan anomali yang

dapat mempengaruhi hasil prediksi (Morgan Kaufmann, 2006).

Data Splitting

Data *Splitting* atau pemisahan data adalah proses membagi data menjadi dua bagian atau lebih data yang memiliki banyak nilai. Data yang sudah diproses dibagi menjadi dua subset: data latih (*training data*) dan data uji (*testing data*). Pembagian ini biasanya menggunakan rasio 80:20, di mana 80% data digunakan untuk melatih model dan 20% sisanya digunakan untuk menguji performa model (Friedman, Hastie, & Tibshirani, 2005).

Pembuatan Model

Dalam tahap ini dilakukan pembuatan model untuk prediksi transaksi perbankan menggunakan metode *Simple Linear Regression*, *Polynomial Regression*, dan *K-Nearest Neighbors (KNN)*, berikut adalah penjelasan dari masing-masing model:

Pembuatan Model *Simple Linear Regression*

Model *Simple Linear Regression* dirancang untuk menentukan hubungan linear antara variabel independen dan dependen (Montgomery, Peck, & Vining, 2021). Model ini diterapkan dengan menggunakan data yang telah dibagi sebelumnya. Tujuannya adalah untuk memprediksi nilai transaksi berdasarkan variabel independen. Model ini akan menghasilkan prediksi nilai transaksi dengan menggunakan garis lurus yang paling sesuai dengan data yang ada. Rumus dari regresi linear sederhana adalah:

$$y = \alpha + bx \dots \text{Persamaan (1)}$$

Keterangan:

- y adalah variabel dependen (yang ingin diprediksi).
- x adalah variabel independen (yang digunakan untuk prediksi).
- α adalah *intercept* (konstanta/titik potong dengan sumbu Y).
- b adalah *slope* (koefisien regresi).

Pembuatan Model *Polynomial Regression*

Model *Polynomial Regression* dikembangkan dengan memperluas model regresi linear sehingga dapat menangkap hubungan non-linear dengan menambahkan derajat polinomial yang lebih tinggi (Draper & Smith, 1998). Pada tahap ini, prediksi nilai

transaksi didasarkan pada variabel independen dengan mempertimbangkan pola yang lebih kompleks. Model ini akan memprediksi nilai transaksi menggunakan kurva polinomial yang paling sesuai dengan data. Rumus dari regresi polinomial adalah:

$$Y = b_0 + b_1X + b_2X^2 + \dots + b_nX^n + \epsilon$$

Persamaan (2)

Keterangan:

- y adalah variabel dependen (yang ingin diprediksi).
- x adalah variabel independen (yang digunakan untuk prediksi).
- $b_0, b_1X, b_2X^2, b_3X^3 \dots b_n$ adalah koefisien regresi.
- ϵ *epsilon* adalah *error term* (residu).
- n adalah derajat polinomial.

Pembuatan Model K-Nearest Neighbors (KNN)

Model *KNN* dibangun dengan menentukan nilai k yang optimal dan menggunakan algoritma ini untuk melakukan prediksi berdasarkan tetangga terdekat (Cover & Hart, 1967). Pada *KNN*, prediksi untuk titik data baru didasarkan pada nilai mayoritas atau rata-rata dari k tetangga terdekat dalam data *training*. Algoritma ini mengukur jarak antara titik data untuk menentukan tetangga terdekat. Rumus dari regresi *K-Nearest Neighbors* dapat dilihat pada persamaan (3) (*KNN*) adalah:

$$\hat{y} = \frac{1}{k} \sum_{i=1}^k y_i \dots \text{Persamaan (3)}$$

Keterangan:

- \hat{y} adalah nilai prediksi untuk data baru
- k adalah jumlah tetangga terdekat yang dipertimbangkan
- y_i adalah nilai dari k tetangga terdekat

Langkah-langkah *KNN*:

1. Tentukan nilai k (jumlah tetangga terdekat).
2. Hitung jarak antara titik data baru dengan semua titik data dalam *dataset*.
3. Pilih k tetangga terdekat berdasarkan jarak.
4. Untuk regresi, hitung rata-rata nilai dari k tetangga terdekat sebagai prediksi.
5. Untuk klasifikasi, gunakan mayoritas kelas dari k tetangga terdekat sebagai prediksi.

Prediksi

Ketiga model yang telah dibuat digunakan untuk melakukan prediksi pada data uji. Perhitungan prediksi dilakukan dengan

mencari nilai y sesuai dengan rumus pada metode yang digunakan. Perhitungan prediksi menggunakan data pengujian hasil dari *splitting* data sebanyak 51 data. Prediksi dapat dilakukan dengan menentukan atribut prediktor (x) untuk dimasukkan ke dalam model yang telah dibuat pada tahap pembuatan model. Hasil prediksi dari setiap model dibandingkan untuk menilai keakuratannya.

Pengujian

Seberapa baik sebuah model dalam meramalkan suatu peristiwa dapat diukur dari besarnya kesalahan antara nilai yang diprediksi dengan nilai yang sebenarnya terjadi. Salah satu ukuran yang sering digunakan untuk mengukur kesalahan prediksi adalah *Root Mean Square Error (RMSE)*. adalah ukuran yang digunakan untuk menilai seberapa akurat suatu model dalam melakukan prediksi. Semakin kecil nilai , semakin baik kinerja model dalam memprediksi data. pada dasarnya adalah rata-rata dari selisih kuadrat antara nilai prediksi dengan nilai sebenarnya Bramasto, S., & Khairiani, D. (2022). Prediksi dianggap sangat akurat jika nilai kurang dari 5. Kriteria dapat dilihat pada tabel berikut:

Tabel 1. Kriteria *RMSE*

Nilai	Kriteria
0 - 5	Sangat Akurat
5 - 10	Akurat
10 - 20	Cukup akurat
> 20	Kurang akurat

Hasil

Pada tahap ini hasil akhir disajikan dalam bentuk tabel dan visualisasi, yang menunjukkan perbandingan kinerja masing-masing model, dimana hal tersebut untuk menentukan metode manakah yang lebih unggul dalam prediksi. Perbandingan yang dimaksud didasarkan pada nilai , model dengan performa terbaik ditentukan berdasarkan nilai yang paling rendah.

3. Hasil dan Pembahasan

Penelitian ini menunjukkan bahwa ketiga metode memiliki performa yang berbeda dalam memprediksi transaksi perbankan. Berikut ini adalah hasil dan pembahasannya:

Pengumpulan Data

Pengumpulan data dilakukan pada 30 November 2024, data diperoleh pada *website*

XYZ. Data yang diambil adalah transaksi bank XYZ dari tanggal 01 Januari 2024 sampai dengan 30 November 2024. Jumlah data yang diperoleh adalah 335 *dataset* yang dapat dilihat pada tabel 2 dibawah ini. Data yang telah diperoleh akan dikumpulkan dan dianalisis agar menjadi informasi yang baru sehingga dapat digunakan untuk memprediksi masalah yang diteliti.

Tabel 2. Isi *Dataset*

No	date	Time	transaction_total
1	01/01/2024	00:00	20925619
2	02/01/2024	00:00	14160416
3	03/01/2024	00:00	21882951
4	04/01/2024	00:00	32264492
5	05/01/2024	00:00	14086672
6	06/01/2024	00:00	15456034
7	07/01/2024	00:00	21848691
8	08/01/2024	00:00	20759711
9	09/01/2024	00:00	16968757
10
331	26/11/2024	00:00	21417317
332	27/11/2024	00:00	24051258
333	28/11/2024	00:00	9759683
334	29/11/2024	00:00	17907877
335	30/11/2024	00:00	17498620
336	30/11/2024	00:00	17498620

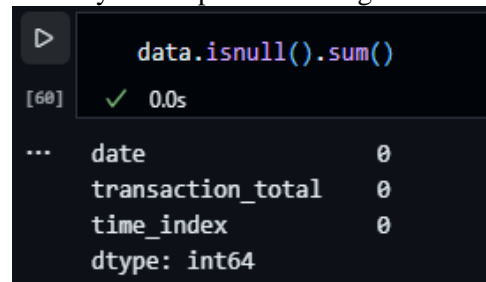
Preprocessing Data

Pada *preprocessing* data akan dilakukan dua proses, yaitu data *selection* dan data *cleaning*. Data yang telah diperoleh mempunyai tiga atribut yaitu *date*, *time*, dan *transaction_total*. Pada tahap data *selection* akan dipilih atribut yang dibutuhkan yaitu atribut *date* dan *transaction_total*. Penyeleksian data dapat dilihat pada tabel 3.

Tabel 3. Hasil data *selection*

No	date	transaction_total
1	01/01/2024	20925619
2	02/01/2024	14160416
3	03/01/2024	21882951
4	04/01/2024	32264492
5	05/01/2024	14086672
6	06/01/2024	15456034
7	07/01/2024	21848691
8	08/01/2024	20759711
9	09/01/2024	16968757
10
331	26/11/2024	21417317
332	27/11/2024	24051258
333	28/11/2024	9759683
334	29/11/2024	17907877
335	30/11/2024	17498620
336	30/11/2024	17498620

Tabel 3 merupakan atribut yang akan digunakan untuk prediksi, yaitu kolom *Date* dan *transaction_total*. Kedua atribut ini saling berkaitan karena nilai pada atribut *transaction_total* dapat bertambah atau berkurang berdasarkan atribut *Date*. Oleh karena itu, kedua atribut tersebut digunakan. Setelah itu, dilakukan proses pembersihan data dengan pengecekan nilai yang hilang pada data. Hasilnya ditampilkan dalam gambar 3.



Gambar 3. Hasil data *cleaning*

Gambar 3 menunjukkan data yang akan digunakan. Tidak ada data yang kosong atau *missing value*, sehingga data siap untuk digunakan pada tahap berikutnya.

Data Splitting

Pada tahap ini, data dipisahkan untuk proses pembuatan model. Variabel independen (x) adalah atribut *time_index*, sedangkan variabel dependen (y) adalah atribut *y_scaled* atau *transaction_totals*.



Gambar 4. *jupyter* untuk memisahkan variabel x dan y

Selanjutnya dilakukan pembagian data menjadi training model dan testing model, Untuk menentukan rasio pembagian antara data *training* dan data *testing*, diperlukan perhitungan akurasi dari kedua metode dengan menggunakan beberapa contoh rasio yang umum digunakan. Beberapa rasio pembagian data ditampilkan pada tabel 4.

Tabel 4. Rasio akurasi metode

Split Ratio	Simple Linear Regression	KNN	Polynomial Regression
80:20	0.18842	0.17394	0.17241
85:15	0.18591	0.16444	0.16925
90:10	0.20693	0.18670	0.19227
Rata-rata	0.19375	0.17503	0.17831

Berdasarkan tabel 4, rasio 85:15 memberikan akurasi tertinggi di antara ketiga metode yang

digunakan untuk pembagian data, dengan akurasi *Simple Linear Regression*, *KNN*, dan *Polynomial Regression*, masing-masing sebesar 0.18591, 0.16444, dan 0.16925. Oleh karena itu, rasio ini akan digunakan dalam proses pembuatan model. Selanjutnya dilakukan proses *split* data, *jupyter* untuk pembagian *training* model dan *testing* model ditampilkan dalam gambar 5.

```

X_train = X[0:284]
y_train = y[0:284]

lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

print("Training samples: (X_train, y_train)")
print("Testing samples: (X_test, y_test)")
    
```

Gambar 5. Pembagian *training* model dan *testing* model

Berdasarkan gambar 5, data dibagi dengan rasio 80:15. Jumlah data *train* adalah 284, sedangkan jumlah data *test* adalah 51. Data *train* akan digunakan untuk pemodelan metode, sementara data *test* digunakan untuk mengukur akurasi metode.

Pembuatan Model

Pada tahap ini model dibuat dengan menggunakan metode *Simple Linear Regression*, *K-Nearest Neighbors (KNN)*, dan *Polynomial Regression*.

Model Simple Linear Regression

Pada tahap ini, model *Simple Linear Regression* dipanggil dan dimasukkan ke dalam variabel linear. Selanjutnya dilakukan proses *training* pada data. Pembuatan model *Simple Linear Regression* dapat dilihat pada gambar 6.

```

lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
    
```

Gambar 6. Pembuatan model *Simple Linear Regression*

Setelah model terbentuk, kita dapat melihat nilai *intercept* (a) dan *slope* (b) yang digunakan sebagai parameter untuk prediksi. *Jupyter* untuk menampilkan nilai *intercept* dan *slope* ditunjukkan dalam gambar 7.

```

intercept_lr = lr_model.intercept_
slope_lr = lr_model.coef_

print("Linear Regression Intercept (a): (intercept_lr[0]-.01)")
print("Linear Regression Slope (b): (slope_lr[0][0]-.01)")
    
```

Gambar 7. Nilai *intercept* dan *slope* *Simple Linear Regression*

Berdasarkan gambar 7, pada model *Simple Linear Regression* yang telah dibuat, nilai *intercept* (a) adalah 0.478945 dan nilai *slope* (b) adalah -0.000086. Sehingga untuk mencari nilai prediksi (y) pada model *Simple Linear Regression* yang telah dibuat adalah dengan memasukan nilai tersebut pada Persamaan (1) sebagai berikut:

$$Y = 0.478945 + (-0.000086x)$$

Model Polynomial Regression

Pada metode *polynomial regression*, variabel independen (X) melalui proses transformasi menggunakan fungsi *polynomial features*. Fungsi ini mengubah data X untuk menciptakan fitur polinomial, yang dapat menangkap hubungan yang lebih kompleks antara variabel independen dan dependen. Setelah transformasi, model regresi linier digunakan untuk melakukan *training* data yang telah diubah (*X_train_poly*) dengan target (*y_train*). Hal ini dilakukan dengan memanggil metode *fit()* pada objek model *Simple Linear Regression()*. *Jupyter* pembuatan model *polynomial regression* dapat dilihat pada gambar 8.

```

poly_model = LinearRegression()
poly_model.fit(X_train_poly, y_train)
    
```

Gambar 8. Pembuatan model *polynomial regression*

Pada Gambar 8 variabel x dilakukan transformasi sebelum proses pembuatan model. Setelah model terbentuk, maka kita dapat melihat nilai *intercept* (b0) dan *slope* atau koefisien-koefisien regresi (b1, b2) yang digunakan sebagai parameter untuk melakukan prediksi. Nilai *intercept* dan *slope* pada *polynomial regression* ditampilkan dalam gambar 9.

```

Intercept = best_poly_model.intercept_
coefficients = best_poly_model.coef_

print(f"Intercept: {Intercept[0]:.4f}")
for i, coef in enumerate(coefficients[0]):
    print(f"Coefficient (Slope) for term {i}: {coef:.4f}")
    
```

Intercept: 0.5105
Coefficient (Slope) for term 0: 0.0000
Coefficient (Slope) for term 1: -0.0008
Coefficient (Slope) for term 2: 0.0000

Gambar 9. Nilai *intercept* dan *slope* *polynomial regression*

Berdasarkan gambar 9, pada model *polynomial regression*, nilai *intercept* (b_0) adalah 0.5105, nilai b_1 adalah -0.0008, dan nilai b_2 adalah 0.0000. Sehingga untuk mencari nilai prediksi (y) pada model *polynomial regression* yang telah dibuat adalah dengan memasukan pada Persamaan (2) sebagai berikut:

$$Y = 0.5105 + (-0.0008x) + 0.0000x^2$$

Model *K-Nearest Neighbors (KNN)*

Pada gambar 10, ditampilkan *jupyter* untuk melatih model *K-Nearest Neighbors (KNN)* dengan menggunakan 6 tetangga terdekat ($n_neighbors=6$), yang kemudian dilatih menggunakan *training* data X_train dan target variabel y_train . Proses ini bertujuan untuk memodelkan hubungan antara variabel *input* (*fitur*) dan *output* (*target*) berdasarkan kedekatan data dengan titik-titik lainnya pada ruang *fitur*.

```

knn_model = KNeighborsRegressor(n_neighbors=6)
knn_model.fit(X_train, y_train)
    
```

KNeighborsRegressor
KNeighborsRegressor(n_neighbors=6)

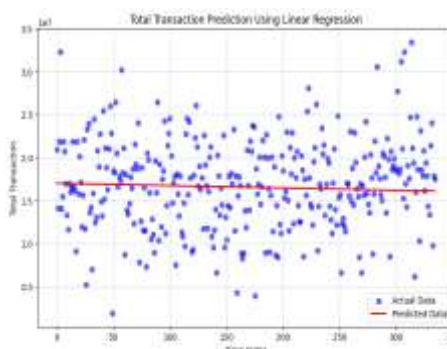
Gambar 10. Pembuatan model *K-Nearest Neighbors (KNN)*

Prediksi

Pada tahap ini dilakukan prediksi dengan menggunakan metode *Simple Linear Regression*, *K-Nearest Neighbors (KNN)*, dan *Polynomial Regression*.

Prediksi *Simple Linear Regression*

Prediksi dilakukan dengan menghitung nilai Y menggunakan Persamaan (1) setelah nilai *intercept* (a) dan *slope* (b) telah diperoleh pada tahap pemodelan. Data yang digunakan untuk menghitung prediksi adalah data *testing*. Data *testing* adalah hasil dari data *split* yang berjumlah 51 data. Visualisasi hasil prediksi menggunakan metode *simple linear regression* (lihat gambar 11).



Gambar 11. Visualisasi model *Simple Linear Regression*

Prediksi *Polynomial Regression*

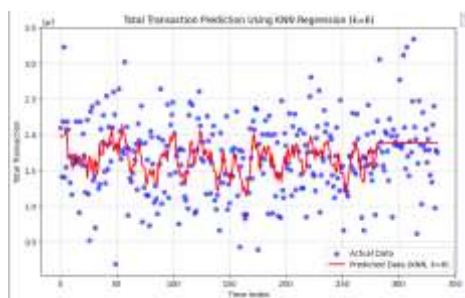
Prediksi dilakukan dengan menghitung nilai Y menggunakan Persamaan (2), setelah nilai *intercept* serta koefisien b_1 dan b_2 diperoleh selama proses pemodelan. Data yang digunakan untuk menghitung prediksi adalah data *testing*, yang merupakan hasil *split* data dan terdiri dari 251 data. Visualisasi hasil prediksi menggunakan metode *polynomial regression* (lihat gambar 12).



Gambar 12. Visualisasi model *polynomial regression*

Prediksi *K-Nearest Neighbors (KNN)*

Prediksi dilakukan dengan mengurutkan data transaksi dari waktu ke waktu. Sumbu Y (*Total Transaction*): Mewakili jumlah total transaksi, dengan skala dari 0 hingga 3.5×10^7 (atau 35 juta). Titik Biru (*Data Aktual*): Menunjukkan data transaksi aktual yang diamati pada indeks waktu tertentu. Garis Merah (*Data Prediksi*): Menunjukkan data transaksi yang diprediksi oleh model regresi *KNN* menggunakan 6 tetangga terdekat ($k=6$). Visualisasi hasil prediksi menggunakan metode *K-Nearest Neighbors (KNN)* (lihat gambar 13).



Gambar 13. Visualisasi model *K-Nearest Neighbors (KNN)*

Pengujian

Pada tahap pengujian, nilai penyimpangan antara nilai aktual dan nilai yang diprediksi sesuai dengan model yang dibuat digunakan sebagai dasar. Metode perhitungan yang digunakan adalah *Root Mean Squared Error (RMSE)*. Nilai dari regresi linear adalah 0.18591, sedangkan nilai dari regresi polinomial adalah 0.16925, dan hasil *KNN* adalah 0.1644.

Hasil Prediksi

Pada pengujian ini, dilakukan prediksi terhadap transaksi Bank XYZ menggunakan tiga metode yang berbeda, yaitu *Simple Linear Regression*, *K-Nearest Neighbors (KNN)*, dan *Polynomial Regression*. Untuk mengevaluasi akurasi model-model tersebut, digunakan *Root Mean Squared Error (RMSE)*, yang mengukur seberapa besar kesalahan rata-rata prediksi model dibandingkan dengan nilai aktual.

Tabel 5. Perbandingan nilai pengujian

Metode	RMSE
<i>Simple Linear Regression</i>	0.18591
<i>K-Nearest Neighbors (KNN)</i>	0.16444
<i>Polynomial Regression</i>	0.16925

Dari hasil pengujian ini, dapat dilihat bahwa model dengan menggunakan *K-Nearest Neighbors (KNN)* menghasilkan nilai yang lebih kecil dibandingkan dengan model lainnya, yaitu 0.16444. Ini menunjukkan bahwa model *KNN* memberikan prediksi yang lebih akurat dibandingkan dengan *Polynomial Regression (RMSE: 0.16925)* dan *Simple*

Linear Regression (RMSE: 0.18591). Dengan hasil tersebut, dapat disimpulkan bahwa semua model yang diuji menunjukkan performa yang sangat baik, karena nilai *RMSE*-nya berada di bawah 5. Hal ini menunjukkan bahwa prediksi 3 model diatas memiliki kesalahan yang relatif kecil dan cukup akurat untuk keperluan prediksi transaksi di Bank XYZ.

4. Kesimpulan

Berdasarkan hasil pengujian yang menggunakan *Root Mean Squared Error (RMSE)*, dapat disimpulkan bahwa *K-Nearest Neighbors (KNN)* merupakan model terbaik untuk memprediksi transaksi Bank XYZ karena memiliki nilai terendah. Meskipun model *Polynomial Regression* juga menunjukkan hasil yang baik, karena lebih rendah dari pada *Simple Linear Regression*, *KNN* tetap menjadi pilihan terbaik dalam hal akurasi prediksi.

Penelitian ini dapat dikembangkan lebih lanjut dengan menggunakan *dataset* yang lebih beragam, termasuk data *real-time*, untuk meningkatkan generalisasi model. Selain itu, peneliti dapat membandingkan metode lain seperti *Random Forest*, *Support Vector Regression (SVR)*, atau *Deep Learning* untuk mengetahui performa metode yang lebih kompleks. Penambahan metrik evaluasi seperti *Mean Absolute Error (MAE)*, *Mean Absolute Percentage Error (MAPE)*, atau *R-squared* akan memberikan gambaran performa model yang lebih komprehensif. Penelitian juga dapat mengintegrasikan variabel eksternal, seperti kondisi ekonomi atau kebijakan perbankan, guna mengukur dampaknya terhadap hasil prediksi. Selain itu, pengembangan sistem prediksi berbasis aplikasi atau web dapat dilakukan untuk implementasi langsung di dunia nyata. Terakhir, uji *robust* terhadap data dengan *outlier* atau ketidakseimbangan juga diperlukan untuk mengevaluasi keandalan model dalam berbagai kondisi data. Pengembangan ini diharapkan mampu meningkatkan akurasi dan relevansi model prediksi transaksi perbankan.

Daftar Pustaka

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), 21-27.

Draper, N. R. (1998). *Applied regression analysis*. McGraw-Hill. Inc.

Montgomery, D. C., Peck, E. A., & Vining, G.

- G. (2021). Introduction to Simple Linear Regression analysis. John Wiley & Sons.
- Sathe, S., & Aggarwal, C. C. (2016, December). Subspace outlier detection in linear time with randomized hashing. In 2016 IEEE 16th International Conference on Data Mining (ICDM) (pp. 459-468). IEEE.
- Mining, W. I. D. (2006). Data mining: Concepts and techniques. Morgan Kaufmann, 10(559-569), 4.
- Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2), 83-85.
- Bramasto, S., & Khairiani, D. (2022). Prediksi daya output sistem Pembangkit Listrik Tenaga Surya (PLTS) menggunakan regresi linear berganda. *Faktor Exacta*, 15(2).
- Satriyo, S. A. L., & Pratama, A. R. (2023). Perbandingan metode linear regresi dan polynomial regresi untuk memprediksi harga saham studi kasus Bank BCA. *INFOTECH: Jurnal Informatika & Teknologi*, 4(1), 59-70.
- Pratiwi, A. A., Saraswati, W. T., Ardiansyah, R. F., Rouf, E. H., & Pratama, A. (2023). Determining The Loan Feasibility of Bank Customers Using Naïve Bayes, K-Nearest Neighbors And Simple Linear Regression Algorithms. *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, 6(3), 226-236.
- Huda, D. M., Dwilestari, G., Rinaldi, A. R., & Solihin, I. (2024). Prediksi Harga Mobil Bekas Menggunakan Algoritma Regresi Linear Berganda. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 6(1), 150-157.
- Fitri, E. (2023). Analisis Perbandingan Metode Regresi Linier, Random Forest Regression dan Gradient Boosted Trees Regression Method untuk Prediksi Harga Rumah. *Journal of Applied Computer Science and Technology*, 4(1), 58-64.
- Nurfauzan, D., & Fatimah, T. (2022, September). Implementasi Algoritma K-Nearest Neighbors Regression Dalam Memprediksi Harga Saham. In *Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)* (Vol. 1, No. 1, pp. 576-584).
- Alfat, L., Hermawan, H., Rustandiputri, A., Inzhagi, R., & Tandjilal, R. (2022). Prediksi Saham PT. Aneka Tambang Tbk. dengan K-Nearest Neighbors. *JSAI (Journal Scientific and Applied Informatics)*, 5(3), 236-243.
- Fauzi, F. A. N., Santoso, R., & Di Asih, I. M. (2022). Pemodelan Data Time Series Menggunakan Pendekatan Regresi Polinomial Lokal Pada Data Harga Saham MDKA. *Indonesian Journal of Applied Statistics*, 6(2), 186-197.
- Rachma, G. Z. N., Suparti, S., & Santoso, R. (2024). PEMODELAN DATA LONGITUDINAL MENGGUNAKAN REGRESI POLINOMIAL LOKAL PADA KELOMPOK SAHAM PERUSAHAAN PENYEDIA JASA TELEKOMUNIKASI DENGAN GUI R. *Jurnal Gaussian*, 12(3), 352-361.
- Rachma, G. Z. N., Suparti, S., & Santoso, R. (2024). PEMODELAN DATA LONGITUDINAL MENGGUNAKAN REGRESI POLINOMIAL LOKAL PADA KELOMPOK SAHAM PERUSAHAAN PENYEDIA JASA TELEKOMUNIKASI DENGAN GUI R. *Jurnal Gaussian*, 12(3), 352-361.
- Priambodo, B., Ahmad, A., & Kadir, R. A. (2021). Spatio-temporal K-NN prediction of traffic state based on statistical features in neighbouring roads. *Journal of Intelligent & Fuzzy Systems*, 40(5), 9059-9072.
- Priambodo, B., Ahmad, A., & Kadir, R. A. (2020). Prediction of average speed based on relationships between neighbouring roads using K-NN and neural network.
- Priambodo, B., Rahayu, S., Naf'an, E., Handriani, I., Putra, Z. P., Nseaf, A. K., ... & Jumaryadi, Y. (2019, December). Predicting GDP of Indonesia Using K-Nearest Neighbour Regression. In *Journal of Physics: Conference Series* (Vol. 1339,

No. 1, p. 012040). IOP Publishing.

Hasanudin, M., Prihandi, I., & Nazua, S. (2024). a Comparative Study of Iconnet Jabodetabek and Banten Using Linear Regression and Support Vector Regression. *Journal of Computer Networks, Architecture and High Performance Computing*, 6(1), 119-

