

# Implementasi Regresi Polinomial untuk Analisis Data Eksperimental

Fathan Yazid Satriani  
Departemen Teknik Elektro  
Universitas Indonesia  
Depok, Indonesia  
NPM: 2306250560

**Abstract**—Regresi polinomial adalah metode statistik yang digunakan untuk memodelkan hubungan antara variabel independen dan variabel dependen sebagai polinomial derajat ke- $n$ . Proyek ini mengimplementasikan algoritma regresi polinomial menggunakan bahasa C++ untuk menganalisis set data eksperimental. Program membaca data dari file, menghitung koefisien polinomial menggunakan metode normal equations yang diselesaikan dengan eliminasi Gauss, dan menyimpan koefisien tersebut. Selanjutnya, skrip Python digunakan untuk memvisualisasikan data asli dan kurva polinomial yang dihasilkan, memberikan representasi grafis dari kecocokan model. Studi kasus ini menunjukkan penerapan regresi polinomial dalam menemukan tren pada data yang mungkin tidak linear secara sederhana. Hasilnya menunjukkan bahwa metode ini efektif dalam mengaproksimasi fungsi yang mendasari data dengan tingkat akurasi yang bergantung pada derajat polinomial yang dipilih dan sifat data itu sendiri.

**Index Terms**—regresi polinomial, metode numerik, pencocokan kurva, C++, Python, analisis data

## I. PENDAHULUAN

Dalam analisis data, seringkali kita dihadapkan pada kebutuhan untuk memahami dan memodelkan hubungan antara berbagai variabel. Salah satu teknik fundamental dalam pemodelan statistik dan pembelajaran mesin adalah regresi. Ketika hubungan antara variabel independen ( $x$ ) dan variabel dependen ( $y$ ) tidak bersifat linear, regresi linear sederhana mungkin tidak cukup untuk menangkap pola yang mendasarinya. Regresi polinomial memperluas konsep regresi linear dengan memodelkan hubungan ini sebagai polinomial derajat ke- $n$ .

Metode ini sangat berguna dalam berbagai bidang ilmu pengetahuan dan teknik, di mana data eksperimental seringkali menunjukkan tren non-linear. Contohnya termasuk pemodelan pertumbuhan populasi, analisis data sensor, prediksi kinerja material, dan banyak lagi. Dengan menyesuaikan derajat polinomial, kita dapat mencapai keseimbangan antara kecocokan model dengan data (goodness-of-fit) dan kompleksitas model untuk menghindari overfitting.

Proyek UAS ini berfokus pada implementasi metode regresi polinomial dari awal menggunakan bahasa pemrograman C++. Tujuannya adalah untuk mengembangkan program yang dapat menerima satu set titik data ( $x, y$ ), menentukan koefisien dari polinomial yang paling sesuai dengan data tersebut untuk derajat yang ditentukan, dan kemudian memvisualisasikan hasilnya. Visualisasi akan dilakukan menggunakan Python dengan pustaka Matplotlib, yang akan menampilkan titik data asli bersama dengan kurva regresi polinomial yang dihitung.

Laporan ini akan menjelaskan secara rinci dasar-dasar teori regresi polinomial, metodologi implementasi dalam C++, data yang digunakan untuk pengujian, analisis hasil eksperimen termasuk plot yang dihasilkan, dan kesimpulan dari proyek ini.

## II. STUDI LITERATUR

Regresi polinomial adalah jenis analisis regresi di mana hubungan antara variabel independen  $x$  dan variabel dependen  $y$  dimodelkan sebagai polinomial derajat  $n$  dalam  $x$ . Persamaan regresi polinomial dapat ditulis sebagai:

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n + \epsilon \quad (1)$$

di mana  $a_0, a_1, \dots, a_n$  adalah koefisien regresi yang perlu diestimasi, dan  $\epsilon$  adalah error acak atau residual. Tujuan dari regresi polinomial adalah untuk menemukan nilai koefisien-koefisien ini sehingga jumlah kuadrat residual (Sum of Squared Residuals, SSR) diminimalkan.

Metode yang umum digunakan untuk menemukan koefisien ini adalah metode kuadrat terkecil (least squares method). Untuk  $m$  titik data  $(x_i, y_i)$ , kita ingin meminimalkan:

$$S = \sum_{i=1}^m (y_i - (a_0 + a_1x_i + a_2x_i^2 + \dots + a_nx_i^n))^2 \quad (2)$$

Untuk meminimalkan  $S$ , kita mengambil turunan parsial dari  $S$  terhadap setiap koefisien  $a_j$  dan menyamakannya dengan nol. Hal ini menghasilkan sistem persamaan linear yang dikenal sebagai *normal equations*. Untuk polinomial derajat  $n$ , sistem ini terdiri dari  $n + 1$  persamaan linear dalam bentuk umum:

$$\begin{aligned} ma_0 + (\sum x_i)a_1 + \dots + (\sum x_i^n)a_n &= \sum y_i \\ (\sum x_i)a_0 + (\sum x_i^2)a_1 + \dots + (\sum x_i^{n+1})a_n &= \sum x_i y_i \\ &\vdots \\ (\sum x_i^n)a_0 + (\sum x_i^{n+1})a_1 + \dots + (\sum x_i^{2n})a_n &= \sum x_i^n y_i \end{aligned} \quad (3)$$

Sistem persamaan linear ini dapat ditulis secara lebih ringkas dalam bentuk matriks sebagai:

$$\mathbf{X}^T \mathbf{X} \mathbf{A} = \mathbf{X}^T \mathbf{Y} \quad (4)$$

di mana  $\mathbf{X}$  adalah matriks desain,  $\mathbf{A}$  adalah vektor koefisien, dan  $\mathbf{Y}$  adalah vektor nilai dependen. Matriks  $\mathbf{X}$  untuk  $m$  data poin dan polinomial derajat  $n$  adalah:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^n \end{bmatrix} \quad (5)$$

Vektor koefisien  $\mathbf{A}$  dan vektor nilai dependen  $\mathbf{Y}$  adalah:

$$\mathbf{A} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad (6)$$

Sistem normal equations (Persamaan 4) kemudian dapat diselesaikan untuk  $\mathbf{A}$  menggunakan berbagai metode aljabar linear, seperti eliminasi Gauss atau dekomposisi LU [1]. Pemilihan derajat polinomial  $n$  adalah aspek penting. Derajat yang terlalu rendah dapat menyebabkan underfitting (model terlalu sederhana), sedangkan derajat yang terlalu tinggi dapat menyebabkan overfitting (model terlalu kompleks dan menangkap noise dalam data) [2].

### III. PENJELASAN DATA YANG DIGUNAKAN

Untuk proyek ini, data yang digunakan adalah data sintetis yang dihasilkan untuk meniru tren polinomial dengan sedikit noise acak. Data ini disimpan dalam file teks bernama `data.txt` di dalam direktori `Program`. Setiap baris dalam file ini merepresentasikan satu titik data  $(x, y)$ , dengan nilai  $x$  dan  $y$  dipisahkan oleh spasi.

Contoh format data dalam `data.txt`:

```
0.0 1.1
0.5 1.8
1.0 2.7
1.5 4.5
2.0 5.8
2.5 8.0
3.0 10.1
3.5 13.5
4.0 17.3
4.5 21.0
5.0 25.5
5.5 30.0
6.0 36.2
```

Data ini dipilih untuk memberikan serangkaian titik yang secara visual menunjukkan tren non-linear, sehingga cocok untuk dianalisis menggunakan regresi polinomial. Program C++ akan membaca data ini untuk melakukan perhitungan regresi. Skrip Python juga akan membaca file data yang sama untuk memplot titik-titik data asli bersama dengan kurva regresi yang dihasilkan.

Penggunaan data sintetis memungkinkan kita untuk memiliki kontrol atas karakteristik data dan mempermudah verifikasi kebenaran implementasi metode numerik. Dalam kasus nyata,

data ini bisa berasal dari berbagai sumber eksperimental atau observasional.

### IV. PENJELASAN METODE YANG DIGUNAKAN

Metode utama yang diimplementasikan dalam proyek ini adalah regresi polinomial menggunakan metode kuadrat terkecil untuk menentukan koefisien polinomial. Algoritma diimplementasikan dalam bahasa C++ dan langkah-langkah utamanya adalah sebagai berikut:

- 1) **Pembacaan Data:** Program C++ pertama-tama membaca titik-titik data  $(x_i, y_i)$  dari file `Program/data.txt`. Setiap titik data disimpan dalam sebuah struktur atau kelas yang sesuai.
- 2) **Penentuan Derajat Polinomial:** Derajat polinomial  $n$  yang akan dicocokkan ditentukan oleh pengguna (dalam kode, ini diatur sebagai variabel, misalnya `'polynomialDegree = 2'` untuk polinomial kuadratik).
- 3) **Pembentukan Normal Equations:** Berdasarkan titik data yang dibaca dan derajat polinomial  $n$ , program membentuk sistem normal equations (Persamaan 4).
  - Matriks  $\mathbf{X}^T \mathbf{X}$  adalah matriks  $(n+1) \times (n+1)$  yang elemennya  $(j, k)$  adalah  $\sum_{i=1}^m x_i^{j+k}$ , di mana  $j, k$  berkisar dari 0 hingga  $n$ .
  - Vektor  $\mathbf{X}^T \mathbf{Y}$  adalah vektor  $(n+1) \times 1$  yang elemennya  $j$  adalah  $\sum_{i=1}^m y_i x_i^j$ .
- 4) **Penyelesaian Sistem Persamaan Linear:** Sistem normal equations yang terbentuk kemudian diselesaikan untuk mendapatkan vektor koefisien  $\mathbf{A} = [a_0, a_1, \dots, a_n]^T$ . Dalam implementasi ini, metode eliminasi Gauss dengan pivoting parsial digunakan untuk menyelesaikan sistem persamaan linear tersebut. Eliminasi Gauss adalah metode standar untuk mengubah matriks augmented menjadi bentuk eselon baris tereduksi, dari mana solusi dapat diperoleh melalui substitusi mundur.
- 5) **Penyimpanan Koefisien:** Setelah koefisien  $a_0, a_1, \dots, a_n$  ditemukan, koefisien tersebut disimpan ke dalam file teks (misalnya, `coefficients_degN.txt`). Setiap koefisien ditulis pada baris baru. File ini disimpan di direktori root proyek.
- 6) **Visualisasi (Python):** Skrip Python `Program/plot_regression.py` kemudian digunakan untuk visualisasi.
  - Skrip membaca kembali titik data asli dari `Program/data.txt`.
  - Skrip membaca koefisien polinomial yang telah dihitung dari file `coefficients_degN.txt` yang berada di direktori root proyek.
  - Menggunakan koefisien ini, skrip menghasilkan nilai  $y$  dari fungsi polinomial  $P(x) = a_0 + a_1x + \dots + a_nx^n$  untuk rentang nilai  $x$  yang sesuai.
  - Akhirnya, skrip memplot titik data asli sebagai scatter plot dan kurva regresi polinomial sebagai line plot menggunakan pustaka Matplotlib. Plot ini disimpan sebagai file gambar (misalnya,

polynomial\_regression\_deg2.png,  
polynomial\_regression\_residuals\_deg2.png,  
polynomial\_regression\_comparative\_deg2.png  
di direktori Laporan.

Komentar ditambahkan dalam kode C++ untuk menjelaskan setiap langkah penting dalam proses implementasi.

## V. DISKUSI DAN ANALISA HASIL EXPERIMEN

Program C++ berhasil dikompilasi dan dijalankan menggunakan data yang disediakan di Program/data.txt. Untuk berbagai derajat polinomial (1, 2, 3, dan 4), program menghitung koefisien polinomial dan menyimpannya di file seperti coefficients\_degN.txt di direktori root proyek.

Skrip Python Program/plot\_regression.py kemudian dijalankan. Skrip ini membaca data asli dan koefisien yang dihasilkan, lalu membuat tiga jenis plot: plot regresi untuk derajat tertentu (default derajat 2), plot residual untuk derajat tersebut, dan plot perbandingan untuk beberapa derajat polinomial (default derajat 1, 2, 3, dan 4).

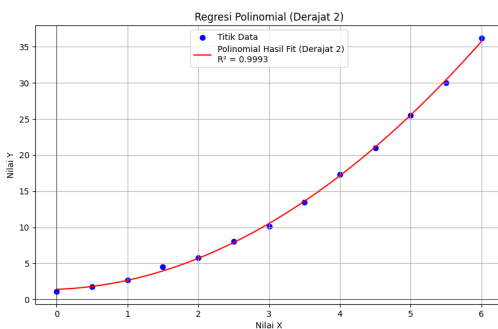


Fig. 1. Hasil Regresi Polinomial (Derajat 2) pada Data Sintetis.

Gambar 1 menunjukkan hasil visualisasi untuk regresi polinomial derajat 2 pada data sampel. Titik-titik biru merepresentasikan data asli dari data.txt, sedangkan garis merah adalah kurva polinomial derajat 2 yang dihitung oleh program C++. Nilai R-squared juga ditampilkan untuk menunjukkan kualitas kecocokan.

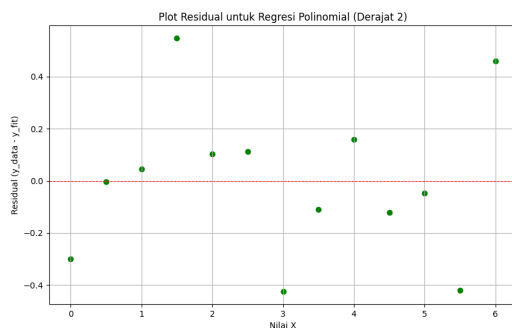


Fig. 2. Plot Residual untuk Regresi Polinomial Derajat 2.

Gambar 2 menampilkan plot residual untuk model regresi polinomial derajat 2. Residual adalah selisih antara nilai y

aktual dan nilai y yang diprediksi oleh model. Idealnya, residual harus tersebar secara acak di sekitar garis nol. Hal ini menunjukkan bahwa model telah menangkap tren utama dalam data dan tidak ada pola sistematis yang tersisa.

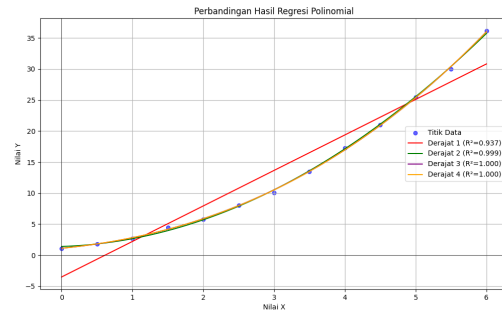


Fig. 3. Perbandingan Hasil Regresi Polinomial untuk Derajat 1, 2, 3, dan 4.

Gambar 3 menyajikan perbandingan kurva regresi untuk derajat polinomial 1, 2, 3, dan 4. Plot ini memungkinkan evaluasi visual tentang bagaimana peningkatan derajat polinomial memengaruhi kecocokan kurva terhadap data. Setiap kurva juga dilengkapi dengan nilai R-squared masing-masing.

### Analisis Hasil:

- **Kecocokan Kurva:** Dari Gambar 1 dan Gambar 3, dapat diamati bahwa kurva polinomial mampu mengikuti tren umum dari titik-titik data. Model regresi polinomial derajat 2 (parabola) tampak memberikan keseimbangan yang baik antara kecocokan dan kompleksitas untuk data sampel ini.
- **Plot Residual:** Plot residual pada Gambar 2 untuk derajat 2 menunjukkan bahwa titik-titik residual tersebar relatif acak di sekitar garis horizontal nol. Meskipun demikian, mungkin ada sedikit pola kurva yang tersisa. Ini bisa mengindikasikan bahwa model derajat 2 mungkin belum sepenuhnya menangkap semua struktur dalam data, atau bisa juga karena sifat dari noise yang ada pada data.
- **Pengaruh Derajat Polinomial:** Gambar 3 secara jelas menunjukkan efek dari perubahan derajat:
  - Derajat 1 (regresi linear) menghasilkan garis lurus. Ini jelas merupakan underfitting untuk data yang melengkung, yang juga ditunjukkan dengan nilai R-squared yang lebih rendah.
  - Derajat 2 dan 3 memberikan kecocokan yang lebih baik. Nilai R-squared meningkat dari derajat 1 ke 2, dan sedikit lagi ke derajat 3.
  - Model derajat 4 memberikan kecocokan yang sangat serupa dengan model derajat 3 untuk data ini. Nilai R-squared mungkin hanya sedikit lebih tinggi, atau bahkan identik. Menaikkan kompleksitas model dari derajat 3 ke 4 bisa jadi tidak menghasilkan perbaikan yang berarti untuk set data ini. Justru, hal ini dapat meningkatkan risiko *overfitting*, terutama jika data mengandung lebih banyak *noise* atau jika jumlah titik data relatif sedikit.

- **Akurasi Metode Numerik:** Akurasi koefisien yang dihitung dipengaruhi oleh stabilitas numerik metode eliminasi Gauss. Selain itu, kondisi matriks normal equations ( $X^T X$ ) juga berperan. Matriks ini dapat menjadi *ill-conditioned* jika derajat polinomial sangat tinggi atau rentang data  $x$  sangat besar. Matriks yang *ill-conditioned* dapat menghasilkan solusi yang kurang akurat. Penggunaan presisi ganda (double) dalam C++ membantu memitigasi masalah ini.
- **Output Program:** Program C++ juga mencetak matriks normal equations dan vektor sisi kanan, serta koefisien yang dihitung ke konsol. Output ini berguna untuk debugging dan verifikasi selama proses pengembangan.

Secara keseluruhan, eksperimen menunjukkan bahwa implementasi regresi polinomial berfungsi sesuai harapan. Kombinasi C++ untuk komputasi numerik inti dan Python untuk visualisasi menyediakan alur kerja yang efektif untuk analisis data semacam ini.

## VI. KESIMPULAN

Proyek ini berhasil mengimplementasikan metode regresi polinomial untuk analisis data eksperimental. Program ini dikembangkan dalam C++ untuk menghitung koefisien polinomial menggunakan metode normal equations yang diselesaikan dengan eliminasi Gauss. Data input dibaca dari file teks, dan koefisien yang dihasilkan disimpan ke file lain. Untuk visualisasi, skrip Python digunakan untuk memplot data asli bersama dengan kurva regresi polinomial yang telah dihitung, memberikan evaluasi visual terhadap kecocokan model.

Dari hasil eksperimen menggunakan data sintetis, dapat disimpulkan bahwa:

- 1) Implementasi regresi polinomial mampu menangkap tren non-linear dalam data secara efektif, sebagaimana ditunjukkan oleh plot yang dihasilkan.
- 2) Pemilihan derajat polinomial adalah faktor kunci yang memengaruhi kualitas kecocokan model. Plot perbandingan dan nilai R-squared membantu dalam memilih derajat yang sesuai untuk menghindari underfitting dan overfitting. Untuk data yang digunakan, derajat 2 atau 3 tampaknya memberikan hasil yang baik.
- 3) Kombinasi C++ untuk komputasi numerik dan Python untuk visualisasi merupakan pendekatan yang kuat dan fleksibel untuk tugas-tugas analisis data numerik.
- 4) Metode numerik yang digunakan, seperti eliminasi Gauss, terbukti cukup untuk menyelesaikan sistem persamaan linear yang muncul dalam konteks regresi ini untuk data dan derajat polinomial yang wajar.

Proyek ini memberikan pemahaman praktis tentang penerapan salah satu metode pencocokan kurva yang fundamental dalam komputasi numerik dan analisis data.

## LINK GITHUB

Repositori Github untuk proyek ini dapat diakses melalui link berikut: <https://github.com/IfanFYS/Proyek-UAS-Komnum>

## LINK YOUTUBE

Video demonstrasi program dapat dilihat di Youtube melalui link berikut: <https://youtu.be/icb7CtffLWc>

## REFERENSI

### REFERENCES

- [1] Chapra, S. C., Canale, R. P. (2015). *Numerical Methods for Engineers* (7th ed.). McGraw-Hill Education. (Terutama Bab 17 untuk Regresi Polinomial dan Bab 9-10 untuk Eliminasi Gauss).
- [2] James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer. (Bab 7 membahas model regresi non-linear termasuk regresi polinomial).
- [3] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). Cambridge University Press.