

Nama : Imam Baihaqqy
NIM : 21120122130078
Mata Kuliah : Metode Numerik
Kelas : D

Berikut adalah penjelasan alur kode, analisis hasil, dan penjabaran untuk masing-masing bagian dari kode yang telah diberikan:

Alur Kode

File 1: “lagrange_interpolation.py”

1. Import Library:
 - “numpy” untuk operasi numerik.
2. Definisi Fungsi “lagrange_interpolation”:
 - Menghitung polinom interpolasi Lagrange untuk suatu nilai “x” berdasarkan data “x_values” dan “y_values”.
 - Fungsi “L(k, x)” digunakan untuk menghitung polinom dasar Lagrange $L_k(x)$.
 - Menggunakan “sum” untuk menjumlahkan hasil perkalian antara nilai “y” dan polinom dasar Lagrange untuk setiap titik data.
3. Blok “if __name__ == “__main__””:
 - Memastikan kode hanya dijalankan ketika file ini dijalankan secara langsung, bukan diimpor sebagai modul.
 - Melakukan pengujian fungsi dengan data contoh.

File 2: “newton_interpolation.py”

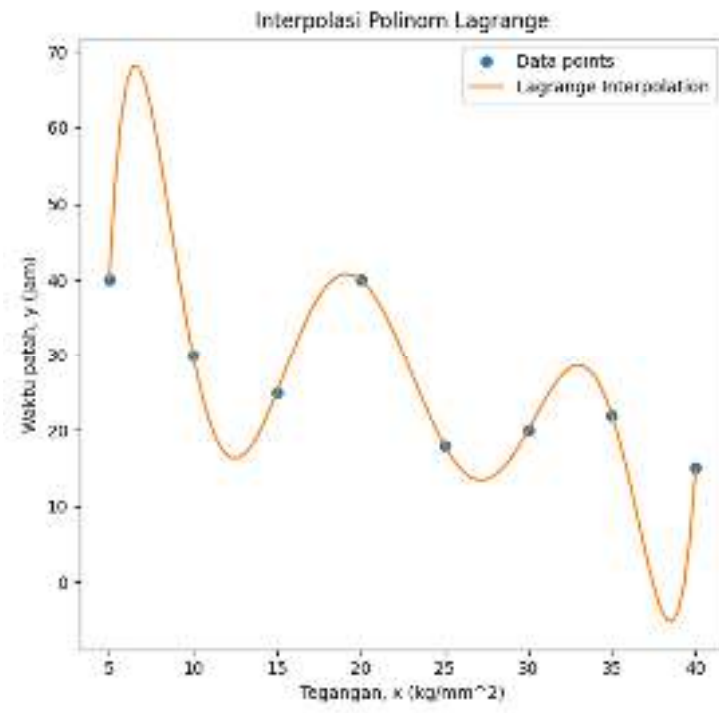
1. Import Library:
 - “numpy” untuk operasi numerik.
2. Definisi Fungsi “newton_interpolation”:
 - Menghitung polinom interpolasi Newton untuk suatu nilai “x” berdasarkan data “x_values” dan “y_values”.
 - Fungsi “divided_differences” digunakan untuk menghitung koefisien perbedaan terbagi Newton.
 - Fungsi “newton_polynomial” digunakan untuk menghitung nilai polinom Newton berdasarkan koefisien yang dihitung.
3. Blok “if __name__ == “__main__””:
 - Sama seperti di “lagrange_interpolation.py”, untuk pengujian langsung.

File 3: “plot_interpolations.py”

1. Import Library:
 - “numpy” untuk operasi numerik.
 - “matplotlib.pyplot” untuk plotting.
 - Mengimpor fungsi “lagrange_interpolation” dan “newton_interpolation” dari file lain.
2. Mempersiapkan Data:
 - Data contoh yang diberikan dalam bentuk array “x_data” dan “y_data”.
3. Membuat Rentang Nilai untuk Plot:
 - “x_plot” adalah rentang nilai dari 5 hingga 40 yang dibagi menjadi 100 titik.
4. Interpolasi Data:
 - Menggunakan fungsi “lagrange_interpolation” dan “newton_interpolation” untuk menghitung nilai interpolasi pada “x_plot”.
5. Plot Hasil Interpolasi:
 - Menggunakan “matplotlib” untuk membuat dua subplot: satu untuk interpolasi Lagrange dan satu untuk interpolasi Newton.
 - Memplot titik data asli sebagai “o” (bulatan) dan hasil interpolasi sebagai garis.
6. Menampilkan Plot:
 - “plt.show()” untuk menampilkan plot yang telah dibuat.

Analisis Hasil

1. Plot Interpolasi Lagrange:

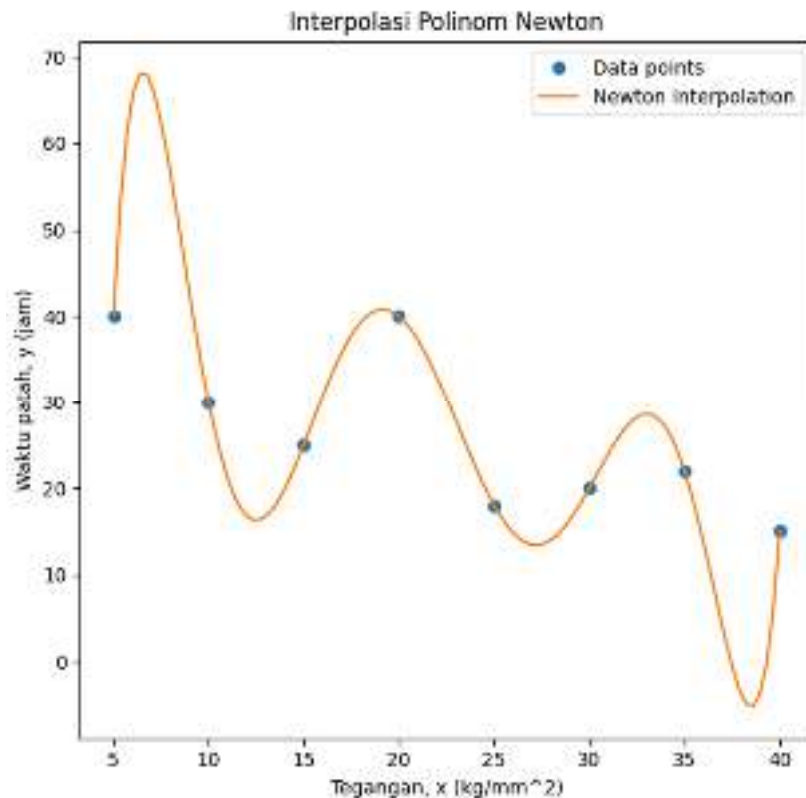


Plot ini menunjukkan polinom Lagrange yang melalui semua titik data. Polinom Lagrange cenderung cocok secara sempurna dengan data yang diberikan, tetapi bisa menjadi sangat osilasi (bergelombang) jika ada banyak titik data atau jika titik-titik tersebut tidak merata.

Penjabaran:

- Interpolasi Polinom Lagrange: Metode ini menghasilkan satu polinom yang melalui semua titik data.
- Kelebihan: Sederhana dan intuitif untuk dipahami.
- Kekurangan: Bisa menjadi tidak stabil dan menghasilkan osilasi yang besar untuk jumlah titik data yang besar atau distribusi titik data yang tidak merata.

2. Plot Interpolasi Newton:



Plot ini menunjukkan polinom Newton yang juga melalui semua titik data. Metode Newton menggunakan perbedaan terbagi yang lebih efisien dalam hal perhitungan, terutama jika Anda menambahkan titik data baru. Polinom Newton umumnya kurang osilasi dibandingkan Lagrange.

Penjabaran:

- Interpolasi Polinom Newton: Metode ini menggunakan perbedaan terbagi untuk menghitung koefisien polinom dan lebih efisien untuk memperbarui polinom saat menambahkan titik data baru.
- Kelebihan: Lebih stabil dibandingkan Lagrange dan lebih efisien untuk memperbarui polinom.
- Kekurangan: Perhitungan koefisien perbedaan terbagi bisa menjadi kompleks untuk implementasi awal.