

Nama : Bravely Dirgayuska

NIM : 21120122140142

Kelas : Metode Numerik C

“TUGAS 1 - IMPLEMENTASI INTERPOLASI”

Soal

Diinginkan aplikasi untuk mencari solusi dari problem pengujian yang memperoleh data terbatas (data terlampir) dengan interpolasi masing-masing menggunakan metode:

1. polinom Langrange

2. polinom Newton

Tugas mahasiswa:

1. Mahasiswa membuat kode sumber dengan bahasa pemrograman yang dikuasai untuk mengimplementasikan solusi di atas

2. Sertakan kode testing untuk menguji kode sumber tersebut untuk menyelesaikan problem dalam gambar. Plot grafik hasil interpolasi dengan $5 \leq x \leq 40$

3. Mengunggah kode sumber tersebut ke Github dan setel sebagai publik. Berikan deskripsi yang memadai dari project tersebut

4. Buat dokumen docx dan pdf yang menjelaskan alur kode dari (1), analisis hasil, dan penjabarannya

- Sebuah pengukuran fisika telah dilakukan untuk menentukan hubungan antara tegangan yang diberikan kepada baja tahan-karat dan waktu yang diperlukan hingga baja tersebut patah. Delapan nilai tegangan yang berbeda dicobakan, dan data yang dihasilkan adalah

Tegangan, x (kg/mm ²)	5	10	15	20	25	30	35	40
Waktu patah, y (jam)	40	30	25	40	18	20	22	15

Jawab :

```
import numpy as np
import matplotlib.pyplot as plt
```

```

# Data tegangan (x) dan waktu patah (y)
x_data = np.array([5, 10, 15, 20, 25, 30, 35, 40])
y_data = np.array([40, 30, 25, 40, 18, 20, 22, 15])

# Fungsi untuk interpolasi Polinom Lagrange
def lagrange_interpolation(x_data, y_data, x):
    def L(k, x):
        term = 1
        for i in range(len(x_data)):
            if i != k:
                term *= (x - x_data[i]) / (x_data[k] - x_data[i])
        return term

    P = 0
    for k in range(len(x_data)):
        P += y_data[k] * L(k, x)
    return P

# Fungsi untuk interpolasi Polinom Newton
def newton_interpolation(x_data, y_data, x):
    n = len(x_data)
    divided_diff = np.zeros((n, n))
    divided_diff[:,0] = y_data

    for j in range(1, n):
        for i in range(n-j):
            divided_diff[i][j] = (divided_diff[i+1][j-1] -
divided_diff[i][j-1]) / (x_data[i+j] - x_data[i])

    def N(x):
        result = divided_diff[0,0]
        product_term = 1
        for i in range(1, n):
            product_term *= (x - x_data[i-1])
            result += divided_diff[0, i] * product_term
        return result

    return N(x)

# Plotting hasil interpolasi

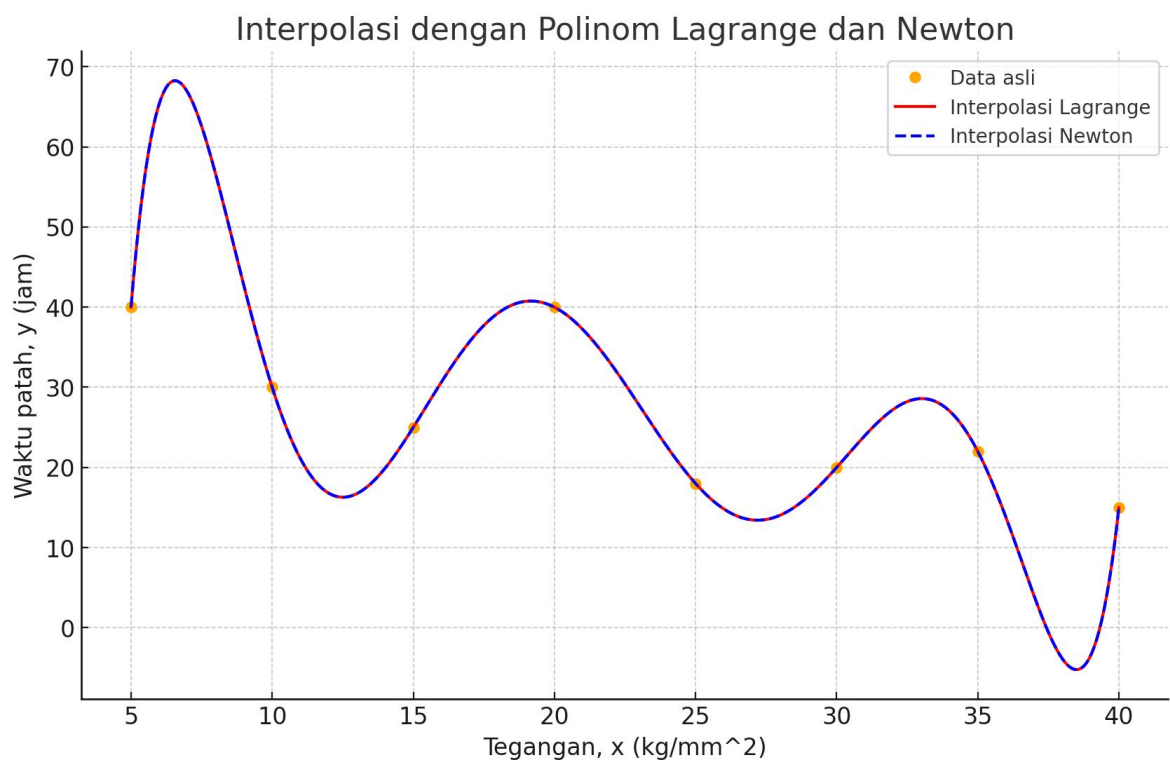
```

```

x_range = np.linspace(5, 40, 400)
y_lagrange = [lagrange_interpolation(x_data, y_data, x) for x in
x_range]
y_newton = [newton_interpolation(x_data, y_data, x) for x in x_range]

plt.plot(x_data, y_data, 'o', label='Data asli', color='orange')
plt.plot(x_range, y_lagrange, label='Interpolasi Lagrange',
color='red')
plt.plot(x_range, y_newton, '--', label='Interpolasi Newton',
color='blue')
plt.xlabel('Tegangan, x (kg/mm^2)')
plt.ylabel('Waktu patah, y (jam)')
plt.legend()
plt.title('Interpolasi dengan Polinom Lagrange dan Newton')
plt.grid(True)
plt.show()

```



Kode tersebut merupakan implementasi metode interpolasi polinomial Lagrange dan Newton untuk memprediksi hubungan antara tegangan (x) dan waktu patah (y) berdasarkan data yang diberikan. Data tegangan dan waktu patah diwakili oleh array numpy `x_data` dan `y_data`.

Fungsi `'lagrange_interpolation'` menggunakan metode Lagrange untuk melakukan interpolasi. Di dalam fungsi ini, terdapat fungsi `'L(k, x)'` yang menghitung basis polinomial Lagrange untuk setiap titik data kecuali titik yang sedang dipertimbangkan (k). Fungsi utama kemudian menggabungkan nilai-nilai basis ini dengan nilai-nilai data y untuk menghasilkan polinom interpolasi yang dievaluasi pada titik x .

Fungsi `'newton_interpolation'` menggunakan metode Newton untuk interpolasi. Ini dimulai dengan membentuk tabel selisih terbagi (`'divided_diff'`) yang diinisialisasi dengan nilai-nilai `y_data` pada kolom pertama. Nilai-nilai selisih terbagi lainnya dihitung secara rekursif. Fungsi `'N(x)'` kemudian menggunakan tabel selisih terbagi ini untuk membentuk dan mengevaluasi polinom interpolasi Newton pada titik x .

Selanjutnya, kode ini membuat plot grafik untuk memvisualisasikan hasil interpolasi. Rentang nilai x antara 5 dan 40 dihasilkan menggunakan `'np.linspace'`, dan fungsi interpolasi Lagrange dan Newton diaplikasikan pada setiap nilai dalam rentang ini untuk mendapatkan `'y_lagrange'` dan `'y_newton'`. Grafik kemudian dibuat menggunakan `'matplotlib'`, dengan data asli ditampilkan sebagai titik oranye, hasil interpolasi Lagrange sebagai garis merah penuh, dan hasil interpolasi Newton sebagai garis putus-putus biru.

Grafik ini menunjukkan bahwa kedua metode interpolasi, baik Lagrange maupun Newton, memberikan hasil yang sangat mirip dan sesuai dengan data asli, menandakan bahwa kedua metode ini efektif untuk memprediksi nilai antara titik-titik data yang diketahui. Perbedaan visual pada hasil interpolasi kedua metode tersebut hampir tidak terlihat, yang menunjukkan bahwa keduanya menghasilkan hasil interpolasi yang hampir identik.