

TUGAS METODE NUMERIK IMPLEMENTASI INTERPOLASI

Nama : Leonardus Deni Prabowo

NIM : 21120122120012

Metode Numerik - Kelas A

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

Source Code

Berikut ini merupakan source code lengkap. Dan di bawahnya merupakan penjelasan dari setiap bagian dari source code yang ada. Source code dibuat dengan menggunakan bahasa Python.

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

# Nama : Leonardus Deni Prabowo
# NIM : 21120122120012
# Metode Numerik - Kelas A
# Tugas Implementasi Interpolasi

# Data Yang Diberi
x_values = np.array([5, 10, 15, 20, 25, 30, 35, 40])
y_values = np.array([40, 30, 25, 40, 18, 20, 22, 15])

# Fungsi Interpolasi Lagrange
def lagrange_interpolation(x, x_points, y_points):
    def L(k, x):
        terms = [(x - x_points[j]) / (x_points[k] - x_points[j]) for j in
range(len(x_points)) if j != k]
        return np.prod(terms)

    result = sum(y_points[k] * L(k, x) for k in range(len(x_points)))
    return result

# Fungsi Interpolasi Newton
def newton_interpolation(x, x_points, y_points):
    def divided_diff(x_points, y_points):
        n = len(y_points)
        coef = np.zeros([n, n])
        coef[:, 0] = y_points

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

        return coef[0, :]

    coeffs = divided_diff(x_points, y_points)
    n = len(coeffs)
    polynomial = coeffs[-1]

    for k in range(1, n):
        polynomial = polynomial * (x - x_points[-k - 1]) + coeffs[-k - 1]

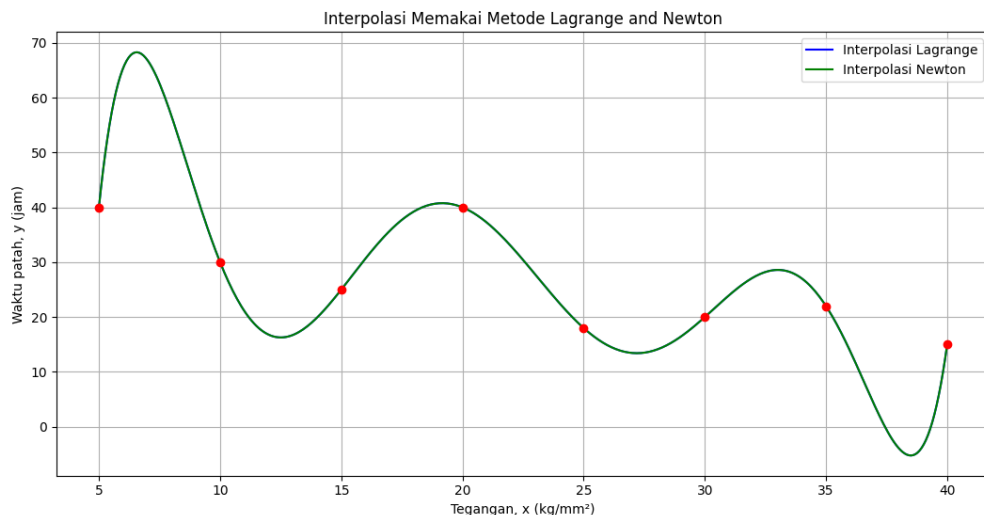
    return polynomial

# Plot Hasil Tabel
x_plot = np.linspace(5, 40, 400)
y_lagrange = [lagrange_interpolation(x, x_values, y_values) for x in
x_plot]
y_newton = [newton_interpolation(x, x_values, y_values) for x in x_plot]

plt.figure(figsize=(12, 6))
plt.plot(x_plot, y_lagrange, label='Interpolasi Lagrange', color='blue')
plt.plot(x_plot, y_newton, label='Interpolasi Newton', color='green')
```

```
plt.scatter(x_values, y_values, color='red', zorder=5)
plt.title('Interpolasi Memakai Metode Lagrange and Newton')
plt.xlabel('Tegangan, x (kg/mm2)')
plt.ylabel('Waktu patah, y (jam)')
plt.legend()
plt.grid(True)
plt.show()
```

Hasil Running



Analisis Program

Berikut ini merupakan analisis dari alur program yang dibuat. Analisis berupa penjelasan tiap-tiap bagian.

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

# Nama : Leonardus Deni Prabowo
# NIM : 21120122120012
# Metode Numerik - Kelas A
# Tugas Implementasi Interpolasi
```

Pada bagian pertama, yang dilakukan adalah memasukkan atau import numpy dan matplotlib.pyplot agar kode dapat berjalan dengan baik. Numpy berfungsi sebagai library yang berfungsi untuk mengelola pengolahan numerik. Dan matplotlib.pyplot berfungsi sebagai visualisasi ketika program dijalankan.

```
# Data Yang Diberi
x_values = np.array([5, 10, 15, 20, 25, 30, 35, 40])
y_values = np.array([40, 30, 25, 40, 18, 20, 22, 15])
```

Pada bagian ini, dimasukkan nilai-nilai sesuai ketentuan yang diberikan. Didefinisikan dua arrah numpy yang berisikan data tegangan (x) dan waktu patah (y). Data ini akan dipakai sebagai titik data untuk interpolasi.

```
# Fungsi Interpolasi Lagrange
def lagrange_interpolation(x, x_points, y_points):
    def L(k, x):
        terms = [(x - x_points[j]) / (x_points[k] - x_points[j]) for j in
range(len(x_points)) if j != k]
        return np.prod(terms)

    result = sum(y_points[k] * L(k, x) for k in range(len(x_points)))
    return result
```

Pada bagian ini merupakan fungsi interpolasi lagrange berfungsi untuk menghitung nilai interpolasi pada titik (x) menggunakan metode polinom langrange. Fungsi 'L(k, x)' di dalamnya akan menghitung basis polinom langrange $(L_k(x))$ dengan cara mengkalikan faktor-faktor yang sesuai. Hasil akhir yang diperoleh dengan melakukan penjumlahan hasil perkalian (y_k) dengan $(L_k(x))$ untuk semua (k) .

```
# Fungsi Interpolasi Newton
def newton_interpolation(x, x_points, y_points):
    def divided_diff(x_points, y_points):
        n = len(y_points)
        coef = np.zeros([n, n])
        coef[:, 0] = y_points

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

        return coef[0, :]

    coeffs = divided_diff(x_points, y_points)
    n = len(coeffs)
    polynomial = coeffs[-1]

    for k in range(1, n):
        polynomial = polynomial * (x - x_points[-k - 1]) + coeffs[-k - 1]

    return polynomial
```

Di bagian ini, terdapat fungsi 'newton_interpolation' menghitung nilai interpolasi pada titik (x) menggunakan metode polinom Newton. Fungsi 'divided_diff' di dalamnya berfungsi untuk melakukan penghitungan tabel selisih terbagi (divided differences), yang kemudian dipakai untuk membentuk sebuah koefisien polinom Newton. Polinom Newton dihitung dilakukan dengan menambahkan kontribusi dari setiap basis polinom.

```
# Plot Hasil Tabel
x_plot = np.linspace(5, 40, 400)
y_lagrange = [lagrange_interpolation(x, x_values, y_values) for x in
x_plot]
y_newton = [newton_interpolation(x, x_values, y_values) for x in x_plot]

plt.figure(figsize=(12, 6))
plt.plot(x_plot, y_lagrange, label='Interpolasi Lagrange', color='blue')
plt.plot(x_plot, y_newton, label='Interpolasi Newton', color='green')
plt.scatter(x_values, y_values, color='red', zorder=5)
plt.title('Interpolasi Memakai Metode Lagrange and Newton')
plt.xlabel('Tegangan, x (kg/mm²)')
```

```
plt.ylabel('Waktu patah, y (jam)')
plt.legend()
plt.grid(True)
plt.show()
```

Pada bagian ini, yang merupakan bagian terakhir, berfungsi sebagai pembuat plot dari hasil interpolasi memakai kedua metode. 'x_plot' adalah array dari 400 titik yang terdistribusi merata di antara 5 dan 40. Nilai interpolasi pada setiap titik akan dihitung memakai fungsi 'lagrange_interpolation' dan 'newton_interpolation', yang disimpan dalam 'y_lagrange' dan 'y_newton'.

Plot akan dibuat memakai matplotlib dengan menampilkan sebuah kurva interpolasi Lagrange yang berwarna biru dan Newton berwarna hijau. Titik data asli akan ditampilkan sebagai scatter plot berwarna merah. Grafik ini akan memberikan visualisasi dari hasil interpolasi. Yang mana memungkinkan perbandingan antara kedua metode yang digunakan dalam mendekati data asli.

Link Github: <https://github.com/LeonDeniP/Tugas-Implementasi-Interpolasi-Leonardus-Deni-Prabowo-21120122120012>