

Metode Numerik  
Tugas Implementasi Interpolasi

Nama : Bagas Farrelino Harsono Putro

NIM : 21120122140101

Kelas : Metode Numerik B

**Implementasi Interpolasi Polinom Langrange dan Newton**

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

# Given data points
x = np.array([5, 10, 15, 20, 25, 30, 35, 40])
y = np.array([40, 30, 25, 40, 18, 20, 22, 15])

# Lagrange interpolation function
def lagrange_interpolation(x_points, y_points, x):
    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, axis=0)

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

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

    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_points[i+j] - x_points[i])

    def N(x):
        result = divided_diff[0, 0]
        product_term = 1.0
        for i in range(1, n):
            product_term *= (x - x_points[i-1])
            result += divided_diff[0, i] * product_term
        return result

    return N(x)

# Generating test points
x_test = np.linspace(5, 40, 500)
```

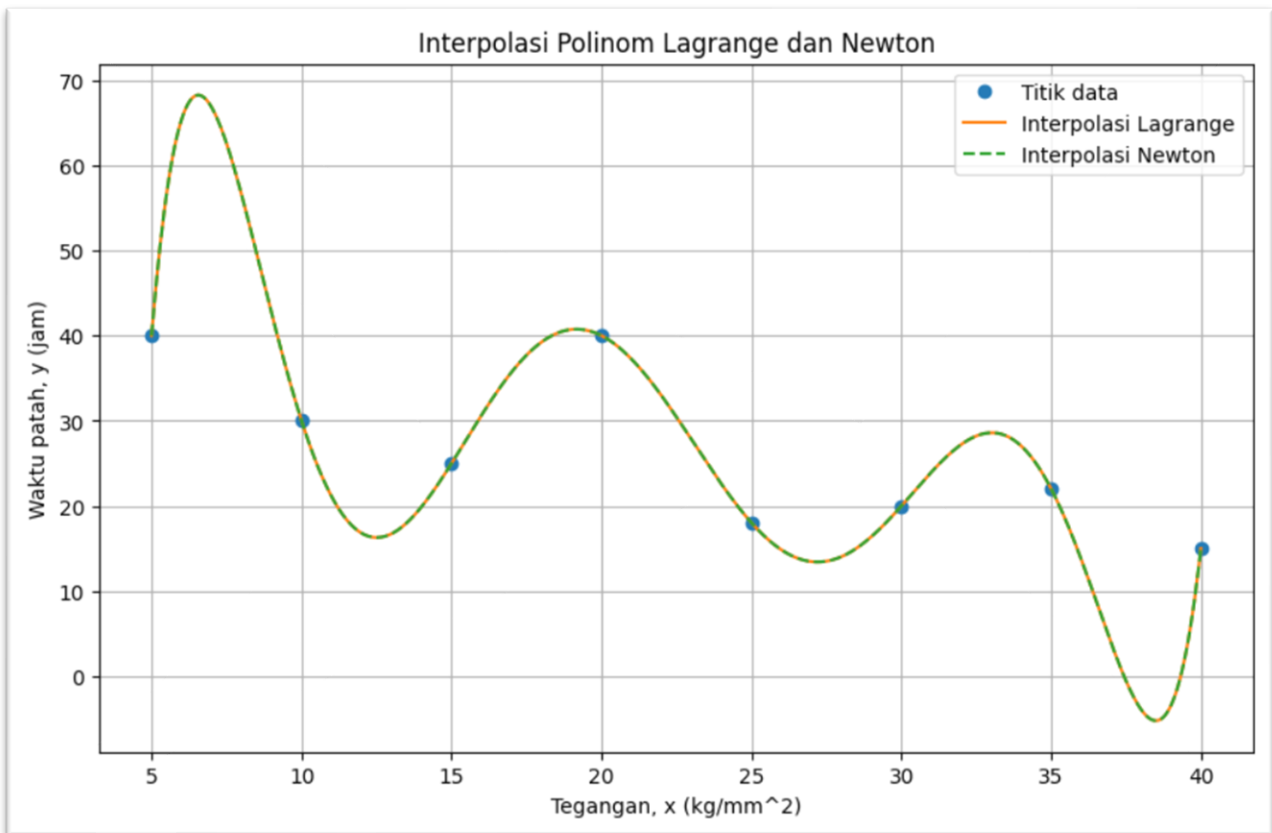
```
# Evaluating the interpolations
y_lagrange = [lagrange_interpolation(x, y, xi) for xi in x_test]
y_newton = [newton_interpolation(x, y, xi) for xi in x_test]

# Plotting the results
plt.figure(figsize=(10, 6))
plt.plot(x, y, 'o', label='Data points')
plt.plot(x_test, y_lagrange, '-', label='Lagrange interpolation')
plt.plot(x_test, y_newton, '--', label='Newton interpolation')
plt.legend()
plt.xlabel('Tegangan, x (kg/mm^2)')
plt.ylabel('Waktu patah, y (jam)')
plt.title('Interpolasi Polinom Lagrange dan Newton')
plt.grid(True)
plt.show()
```

## Alur Kode

1. Data Input : Data titik-titik yang diberikan disimpan dalam array `x` dan `y`.
2. Fungsi Interpolasi : Dua fungsi interpolasi didefinisikan, yaitu `lagrange\_interpolation` dan `newton\_interpolation`.
  - Fungsi `lagrange\_interpolation` menghitung nilai interpolasi menggunakan metode Lagrange
  - Fungsi `newton\_interpolation` menghitung nilai interpolasi menggunakan metode Newton dengan diferensiasi terbagi
3. Pembuatan titik uji: Sebuah array titik uji (`x\_test`) dibuat untuk mengevaluasi interpolasi pada rentang yang lebih halus.
4. Evaluasi interpolasi: Kedua metode interpolasi dievaluasi pada setiap titik uji, menghasilkan nilai-nilai `y` yang sesuai.
5. Plotting: Hasil interpolasi dari kedua metode ditampilkan dalam sebuah plot menggunakan Matplotlib.
  - Data asli ditandai sebagai titik-titik
  - Interpolasi Lagrange ditampilkan sebagai garis solid
  - Interpolasi Newton ditampilkan sebagai garis putus-putus
6. Penyertaan label dan judul: Label sumbu dan judul plot ditambahkan
7. Tampilan plot: Plot ditampilkan

## Hasil Grafik



## Analisis Hasil

Grafik di atas menunjukkan hasil interpolasi data menggunakan metode polinomial Lagrange dan Newton, dengan sumbu x mewakili tegangan (kg/mm<sup>2</sup>) dan sumbu y mewakili waktu patah (jam). Titik data yang diamati ditandai dengan titik biru, sementara hasil interpolasi Lagrange dan Newton masing-masing ditampilkan dengan garis oranye dan garis putus-putus hijau.

Kedua metode interpolasi menghasilkan kurva yang hampir identik, menunjukkan bahwa kedua metode tersebut memberikan hasil interpolasi yang sangat mirip untuk dataset ini. Pola kurva yang kompleks mencerminkan variasi non-linear dalam data, yang ditangkap dengan baik oleh kedua metode interpolasi. Grafik ini mengindikasikan bahwa baik interpolasi polinomial Lagrange maupun Newton dapat digunakan secara efektif untuk memperkirakan nilai di antara titik-titik data yang ada, yang berguna dalam berbagai aplikasi teknik dan ilmiah.