

Nama : Listyawan Femil Anaki

NIM : 21120122140109

Mata Kuliah : Metode Numerik – B

1. Implementasi Interpolasi Polinom Lagrange dan Newton

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

# Data points
data_points = {
    5: 40,
    10: 30,
    15: 25,
    20: 40,
    25: 18,
    30: 20,
    35: 22,
    40: 15
}

# Implementasi Polinom Lagrange
def lagrange_interpolation(x, data_points):
    result = 0
    for xi in data_points:
        term = data_points[xi]
        for xj in data_points:
            if xi != xj:
                term *= (x - xj) / (xi - xj)
        result += term
    return result

# Implementasi Polinom Newton
def newton_interpolation(x, data_points):
    n = len(data_points)
    x_points = list(data_points.keys())
    y_points = list(data_points.values())

    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])

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

    return result

# Testing Kode Sumber dan Plot Grafik
x_values = np.linspace(5, 40, 100)
y_lagrange = [lagrange_interpolation(x, data_points) for x in
               x_values]
y_newton = [newton_interpolation(x, data_points) for x in
            x_values]

# Plotting results for Lagrange
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(x_values, y_lagrange, label='Lagrange Interpolation')
plt.scatter(data_points.keys(), data_points.values(),
            color='green', zorder=5)
plt.legend()
plt.xlabel('Tegangan (kg/mm^2)')
plt.ylabel('Waktu Patah (jam)')
plt.title('Interpolasi Polinom Lagrange')

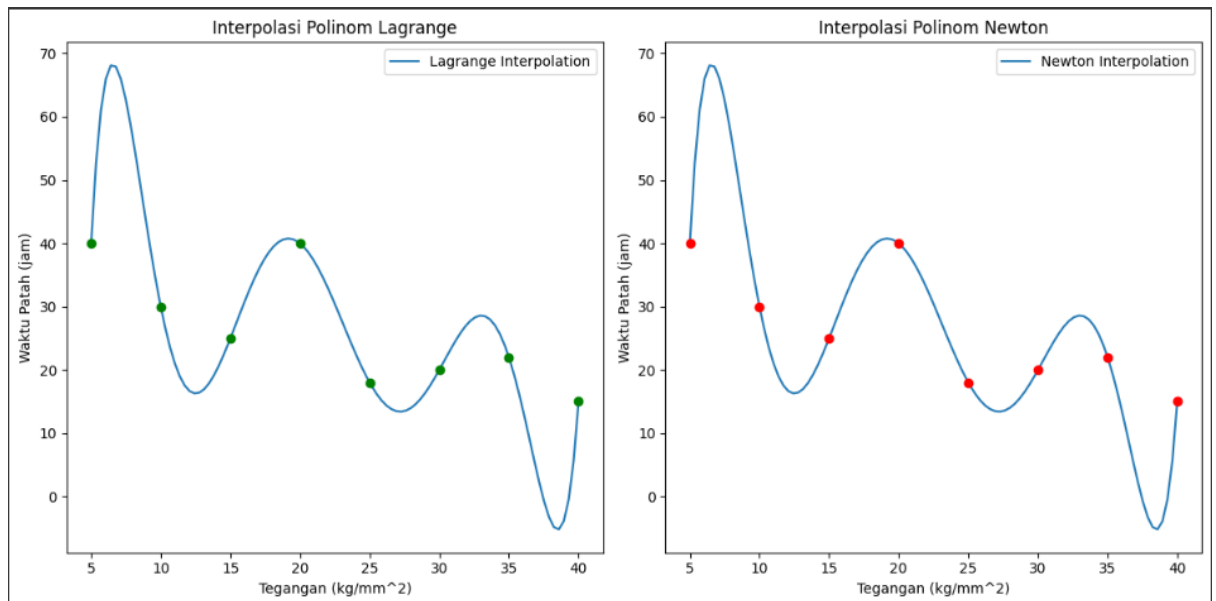
# Plotting results for Newton
plt.subplot(1, 2, 2)
plt.plot(x_values, y_newton, label='Newton Interpolation')
plt.scatter(data_points.keys(), data_points.values(),
            color='red', zorder=5)
plt.legend()
plt.xlabel('Tegangan (kg/mm^2)')
plt.ylabel('Waktu Patah (jam)')
plt.title('Interpolasi Polinom Newton')

plt.tight_layout()

```

```
plt.show()
```

2. Hasil Grafik



3. Penjelasan Alur Kode

1. Data Points

- Kode dimulai dengan mendefinisikan `data_points`, yang merupakan dictionary yang berisi pasangan nilai `x` dan `y` untuk titik-titik data yang akan diinterpolasi.

2. Fungsi Lagrange Interpolation

- `lagrange_interpolation(x, data_points)` adalah fungsi yang mengimplementasikan interpolasi polinomial Lagrange.
- Fungsi ini mengiterasi melalui setiap titik data x_i dan menghitung term $L(x_i)$ menggunakan rumus interpolasi Lagrange.

3. Fungsi Newton Interpolation

- `newton_interpolation(x, data_points)` adalah fungsi yang mengimplementasikan interpolasi polinomial Newton.
- Fungsi ini menggunakan tabel diferensi terbagi untuk menghitung koefisien-koefisien polinomial Newton, dan kemudian menghitung nilai interpolasi untuk nilai `x` yang diberikan.

4. Testing dan Plotting

- Kode menghasilkan `x_values` sebagai array nilai `x` untuk plotting dan menghitung `y_lagrange` dan `y_newton` sebagai hasil interpolasi untuk setiap nilai `x_values`.
- Kemudian, dilakukan plotting menggunakan Matplotlib untuk memvisualisasikan hasil interpolasi.
- Dua grafik ditampilkan secara berdampingan:
 - Grafik kiri menunjukkan hasil interpolasi polinomial Lagrange.
 - Grafik kanan menunjukkan hasil interpolasi polinomial Newton.
 - Titik-titik data asli ditampilkan sebagai titik merah, sedangkan kurva biru mewakili hasil interpolasi.

4. Analisis Grafik

1. Interpolasi Polinomial Lagrange

Grafik pada subplot kiri menampilkan hasil interpolasi menggunakan metode polinomial Lagrange. Kurva biru mengikuti pola yang melewati semua titik data yang diberikan dengan baik. Namun, interpolasi ini cenderung lebih "bergelombang" atau memiliki bentuk yang meliuk-luk dengan lekukan yang mengikuti distribusi titik data. Hal ini terjadi karena interpolasi Lagrange memastikan untuk melewati setiap titik data secara eksak, tanpa menghiraukan bentuk keseluruhan fungsi yang dihasilkan.

2. Interpolasi Polinomial Newton

Grafik pada subplot kanan menunjukkan hasil interpolasi menggunakan metode polinomial Newton. Kurva biru juga melewati semua titik data dengan baik, tetapi bentuknya lebih "halus" dibandingkan dengan interpolasi Lagrange. Polinomial Newton menggunakan diferensi terbagi untuk mengekstrapolasi data, yang menghasilkan bentuk kurva yang lebih terstruktur dan halus.