

Nama : Dimas Gary Irawan

NIM : 21120122140164

Metode Numerik – D

Teknik Komputer / 2022

Github : https://github.com/garyirawan/Implementasi-Interpolasi_Dimas-Gary-Irawan_21120122140164

TUGAS 2 – IMPLEMENTASI INTERPOLASI

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

1. Polinom Lagrange
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 $5 \leq x \leq 40$.

JAWAB:

A. Polinom Lagrange

Interpolasi polinom Lagrange adalah suatu metode untuk menemukan sebuah polinom yang melewati sekumpulan titik data tertentu. Polinom ini digunakan dalam interpolasi, yang bertujuan untuk memperkirakan nilai fungsi di antara titik-titik data yang diketahui.

Di mana

$$P_j(x) = y_j \prod_{\substack{k=1 \\ k \neq j}}^n \frac{x - x_k}{x_j - x_k}.$$

Ditulis secara eksplisit,

$$P(x) = \frac{(x - x_2)(x - x_3) \cdots (x - x_n)}{(x_1 - x_2)(x_1 - x_3) \cdots (x_1 - x_n)} y_1 + \frac{(x - x_1)(x - x_3) \cdots (x - x_n)}{(x_2 - x_1)(x_2 - x_3) \cdots (x_2 - x_n)} y_2 + \cdots + \frac{(x - x_1)(x - x_2) \cdots (x - x_{n-1})}{(x_n - x_1)(x_n - x_2) \cdots (x_n - x_{n-1})} y_n.$$

Gambar 1 Rumus Interpolasi Lagrange

Sumber: <https://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html>

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

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

def lagrange_interpolation(x_values, y_values, x_point):
    def basis(j):
        p = [(x_point - x_values[m]) / (x_values[j]
        - x_values[m]) for m in range(len(x_values)) if m !=
        j]
        return np.prod(p, axis=0)

    return sum(y_values[j] * basis(j) for j in
    range(len(x_values)))

x_plot = np.linspace(5, 40, 100)
y_plot_lagrange = [lagrange_interpolation(x, y, xi) for xi
in x_plot]

def test_lagrange_interpolation():
    test_x = np.array([7, 12, 18, 28, 33, 38])
    test_y = [lagrange_interpolation(x, y, xi) for xi
in test_x]
    return test_x, test_y

test_x, test_y = test_lagrange_interpolation()

plt.plot(x_plot, y_plot_lagrange, label="Interpolasi
Lagrange")
plt.scatter(x, y, color='red', label='Titik Data')
plt.scatter(test_x, test_y, color='blue', label='Titik
Interpolasi', marker='x')
plt.xlabel('Tegangan, x (kg/mm^2)')
plt.ylabel('Waktu patah, y (jam)')
plt.title('Interpolasi Polinom Lagrange')
```

```
plt.legend()
plt.grid(True)
plt.show()

for i, xi in enumerate(test_x):
    print(f'Interpolasi pada x = {xi}: y = {test_y[i]}')
```

Langkah-langkah dari kode interpolasi polinomial lagrange yang digunakan:

1. Import Library:

'numpy' digunakan untuk operasi numerik pada array dan 'matplotlib.pyplot' digunakan untuk visualisasi data.

2. Mendefinisikan Data:

'x' dan 'y' adalah array yang berisi titik data yang diketahui.

3. Definisi Fungsi Interpolasi Lagrange:

'lagrange_interpolation' adalah fungsi untuk menghitung nilai interpolasi di 'x_point' menggunakan metode Lagrange. Fungsi 'basis(j)' menghitung nilai basis polinom Lagrange untuk indeks j. 'np.prod(p, axis=0)' menghitung hasil kali dari elemen-elemen dalam list p. 'sum(y_values[j] * basis(j) for j in range(len(x_values)))' menghitung jumlah kontribusi dari setiap basis polinom yang dikalikan dengan nilai y yang sesuai.

4. Membuat Data untuk Plot:

'x_plot' adalah array yang berisi 100 titik yang terdistribusi merata antara 5 dan 40. 'y_plot_lagrange' adalah nilai interpolasi untuk setiap 'xi' dalam 'x_plot'.

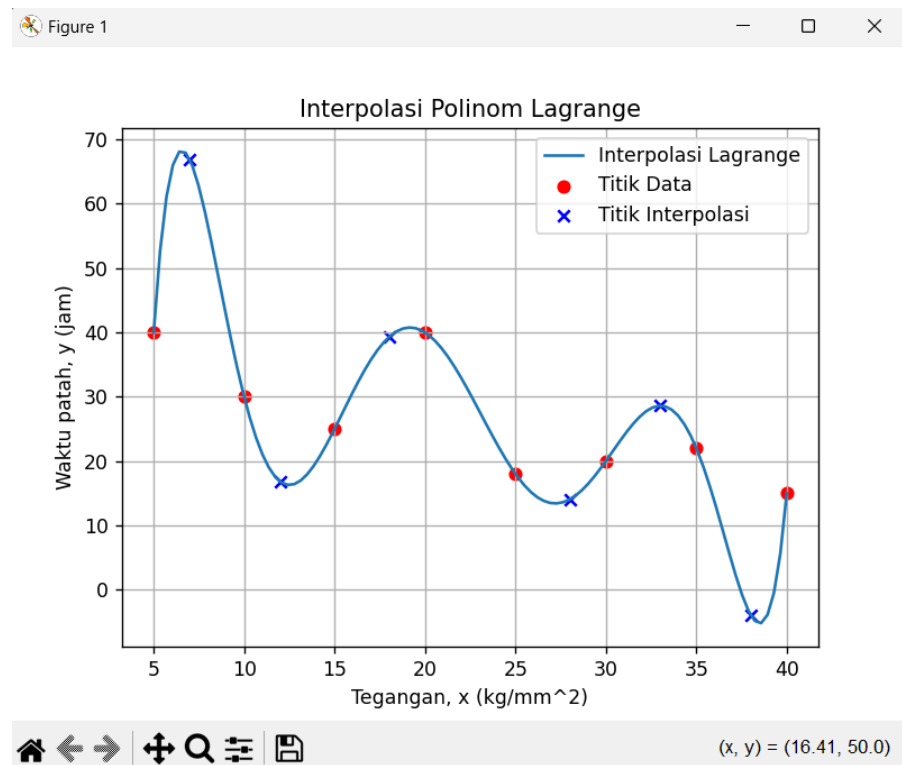
5. Fungsi Pengujian Interpolasi:

'test_lagrange_interpolation' adalah fungsi untuk menguji interpolasi pada beberapa titik yang diberikan '(test_x)'.

6. Plot Hasil Interpolasi:

'plt.plot' untuk menggambar kurva interpolasi. 'plt.scatter' untuk menampilkan titik data asli (x, y) dan titik interpolasi ('test_x', 'test_y'). Label dan judul ditambahkan untuk memperjelas plot.

7. Hasil Interpolasi:



Gambar 2 Hasil Interpolasi Polinom Lagrange

Pada gambar di atas menunjukkan hasil dari interpolasi polinom lagrange dengan data yang ada (terlampir). Terdapat 2 sumbu yaitu sumbu x dan sumbu y. Sumbu X menunjukkan data tegangan yang diukur dalam satuan kg/mm^2 . Sedangkan sumbu Y menunjukkan data waktu patah yang diukur dalam satuan jam. Kemudian terdapat garis biru melengkung yang menggambarkan hasil dari interpolasi polinomial Lagrange yang dihitung untuk rentang nilai x dari 5 hingga 40.

Dalam gambar di atas terdapat dua titik sebagai penanda titik data dan titik interpolasi. Titik merah menunjukkan titik data yang diketahui. Titik-titik tersebut yang digunakan untuk membangun polinomial Lagrange. Sedangkan Titik biru (berbentuk 'x') menunjukkan titik-titik yang diinterpolasi menggunakan metode polinomial Lagrange.

B. Polinom Newton

Interpolasi polinom Newton adalah metode lain untuk menemukan sebuah polinom yang melewati sekumpulan titik data tertentu. Polinom Newton digunakan untuk memperkirakan nilai fungsi di antara titik-titik data yang diketahui. Metode ini sangat mirip dengan interpolasi polinom Lagrange tetapi memiliki bentuk dan pendekatan yang berbeda dalam pembangunannya.

Rumus Umum

Rumus umum yang sering digunakan untuk interpolasi polinomial newton adalah.

$$f_{n-1}(x) = b_1 + b_2(x-x_1) + b_3(x-x_1)(x-x_2) + \dots + b_n(x-x_1)(x-x_2) \dots (x-x_{n-1})$$

Nilai b_1 dapat diperoleh dengan cara memberikan nilai $x = x_1$ sehingga $x - x_1 = 0$ dan dengan demikian diperoleh

$$b_1 = f(x_1)$$

Nilai b_2, b_3, b_4 dan seterusnya dapat diperoleh dengan beda bagi, yaitu

$$b_2 = f[x_2, x_1]$$

$$b_3 = f[x_3, x_2, x_1]$$

$$b_4 = f[x_4, x_3, x_2, x_1]$$

Gambar 3 Rumus Interpolasi Newton

Sumber: <https://www.rahmateduc.com/2019/08/interpolasi-newton.html>

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

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

def divided_diff(x_values, y_values):
    n = len(y_values)
    coef = np.zeros([n, n])
    coef[:, 0] = y_values

    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_values[i + j] - x_values[i])

    return coef[0, :]

def newton_interpolation(x_values, y_values, x_point):
    coef = divided_diff(x_values, y_values)
    n = len(coef)
    y_interp = coef[0]
    for i in range(1, n):
```

```

        y_interp += coef[i] * np.prod([x_point -
x_values[j] for j in range(i)])
    return y_interp

x_plot = np.linspace(5, 40, 100)
y_plot_newton = [newton_interpolation(x, y, xi) for xi in
x_plot]

def test_newton_interpolation():
    test_x = np.array([7, 12, 18, 28, 33, 38])
    test_y = [newton_interpolation(x, y, xi) for xi in
test_x]
    return test_x, test_y

test_x, test_y = test_newton_interpolation()

plt.plot(x_plot, y_plot_newton, label="Interpolasi
Newton")
plt.scatter(x, y, color='red', label='Titik Data')
plt.scatter(test_x, test_y, color='blue', label='Titik
Interpolasi', marker='x')
plt.xlabel('Tegangan, x (kg/mm^2)')
plt.ylabel('Waktu patah, y (jam)')
plt.title('Interpolasi Polinom Newton')
plt.legend()
plt.grid(True)
plt.show()

for i, xi in enumerate(test_x):
    print(f'Interpolasi pada x = {xi}: y = {test_y[i]}')

```

Langkah-langkah dari kode interpolasi polynomial newton yang digunakan:

1. Import Library:

'numpy' digunakan untuk operasi numerik pada array dan 'matplotlib.pyplot' digunakan untuk visualisasi data.

2. Mendefinisikan Data:

'x' dan 'y' adalah array yang berisi titik data yang diketahui.

3. Fungsi Divided Difference:

'divided_diff' adalah fungsi untuk menghitung tabel selisih terbagi (divided difference table) yang menghasilkan koefisien polinomial Newton. 'n' adalah jumlah titik data. coef adalah matriks yang menyimpan koefisien selisih terbagi. Dua loop for mengisi matriks 'coef' dengan selisih terbagi hingga menghasilkan baris pertama dari tabel selisih terbagi, yaitu koefisien untuk polinomial Newton.

4. Fungsi Interpolasi Newton

`'newton_interpolation'` adalah fungsi untuk menghitung nilai interpolasi pada `'x_point'` menggunakan polinomial Newton. `'coef'` adalah koefisien yang dihitung oleh fungsi `'divided_diff'`. `'y_interp'` mulai dengan nilai koefisien pertama. Loop for menambahkan kontribusi dari setiap basis polinomial Newton, dikalikan dengan koefisien yang sesuai.

5. Membuat Data untuk Plot:

`'x_plot'` adalah array yang berisi 100 titik yang terdistribusi merata antara 5 dan 40. `'y_plot_lagrange'` adalah nilai interpolasi untuk setiap `'xi'` dalam `'x_plot'`.

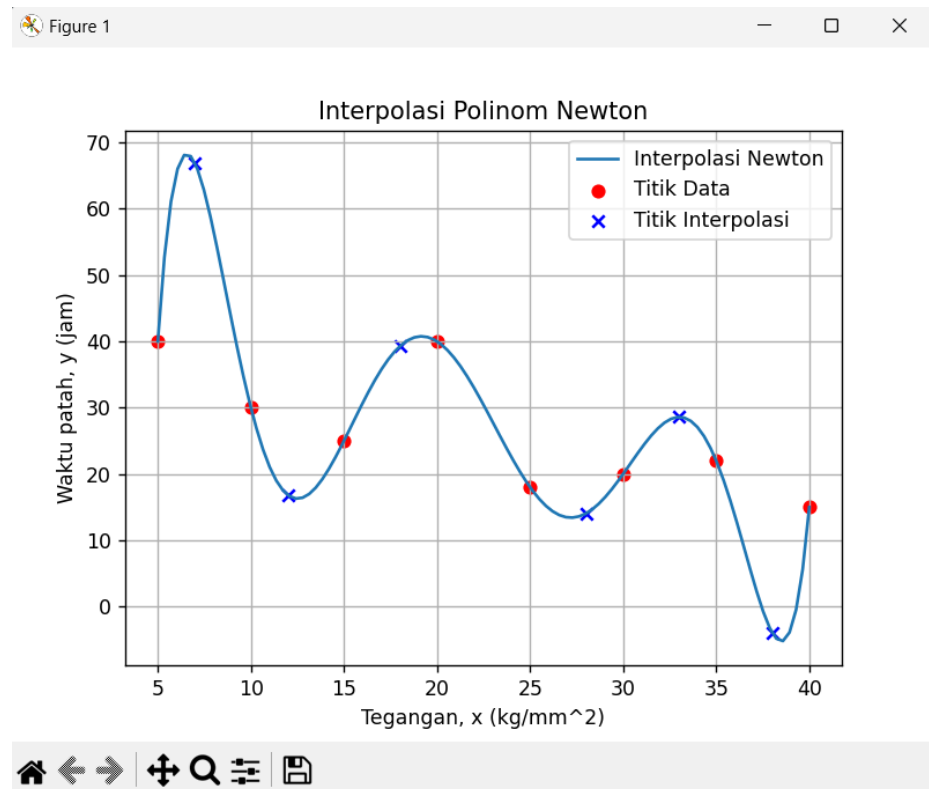
6. Fungsi Pengujian Interpolasi:

`'test_newton_interpolation'` adalah fungsi untuk menguji interpolasi pada beberapa titik yang diberikan (`'test_x'`).

7. Plot Hasil Interpolasi:

`'plt.plot'` untuk menggambar kurva interpolasi. `'plt.scatter'` untuk menampilkan titik data asli (x, y) dan titik interpolasi (`'test_x'`, `'test_y'`). Label dan judul ditambahkan untuk memperjelas plot.

8. Hasil Interpolasi



Gambar 4 Hasil Interpolasi Polinom Lagrange

Pada gambar di atas menunjukkan hasil dari interpolasi polinom newton dengan data yang ada (terlampir). Terdapat 2 sumbu yaitu sumbu x dan sumbu y. Sumbu X menunjukkan data tegangan yang diukur dalam satuan kg/mm². Sedangkan sumbu Y menunjukkan data waktu patah yang diukur dalam satuan jam. Kemudian terdapat garis biru melengkung yang menggambarkan hasil dari interpolasi polinomial newton yang dihitung untuk rentang nilai x dari 5 hingga 40.

Dalam gambar di atas terdapat dua titik sebagai penanda titik data dan titik interpolasi. Titik merah menunjukkan titik data yang diketahui. Titik-titik tersebut yang digunakan untuk membangun polynomial newton. Sedangkan Titik biru (berbentuk 'x') menunjukkan titik-titik yang diinterpolasi menggunakan metode polynomial newton.