

Nama : Efesus G.K. Banurea
NIM : 21120122140117
Jurusan : Teknik Komputer
Mata Kuliah : Metode Numerik (D)

Interpolasi adalah metode matematika yang digunakan untuk memperkirakan nilai di antara dua titik data yang diketahui. Dalam tugas ini, kita menggunakan dua metode interpolasi: polinom Lagrange dan polinom Newton, untuk menghitung nilai perkiraan berdasarkan data tegangan dan waktu patah yang diberikan.

Alur Kode

1. Interpolasi Polinom Lagrange

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

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

# Fungsi untuk interpolasi Lagrange
def lagrange_interpolation(x_values, y_values, x):
    def basis(i, x):
        term = [(x - x_values[j]) / (x_values[i] -
x_values[j]) for j in range(len(x_values)) if j != i]
        return np.prod(term, axis=0)
```

```

        return sum(y_values[i] * basis(i, x) for i in
range(len(x_values)))

# Plotting hasil interpolasi

x_range = np.linspace(5, 40, 400)

y_lagrange = [lagrange_interpolation(x, y, xi) for xi in
x_range]

plt.figure(figsize=(12, 6))

plt.plot(x, y, 'o', label='Data Asli')

plt.plot(x_range, y_lagrange, label='Interpolasi
Lagrange')

plt.xlabel('Tegangan, x (kg/mm^2)')

plt.ylabel('Waktu Patah, y (jam)')

plt.legend()

plt.title('Interpolasi dengan Polinom Lagrange')

plt.grid(True)

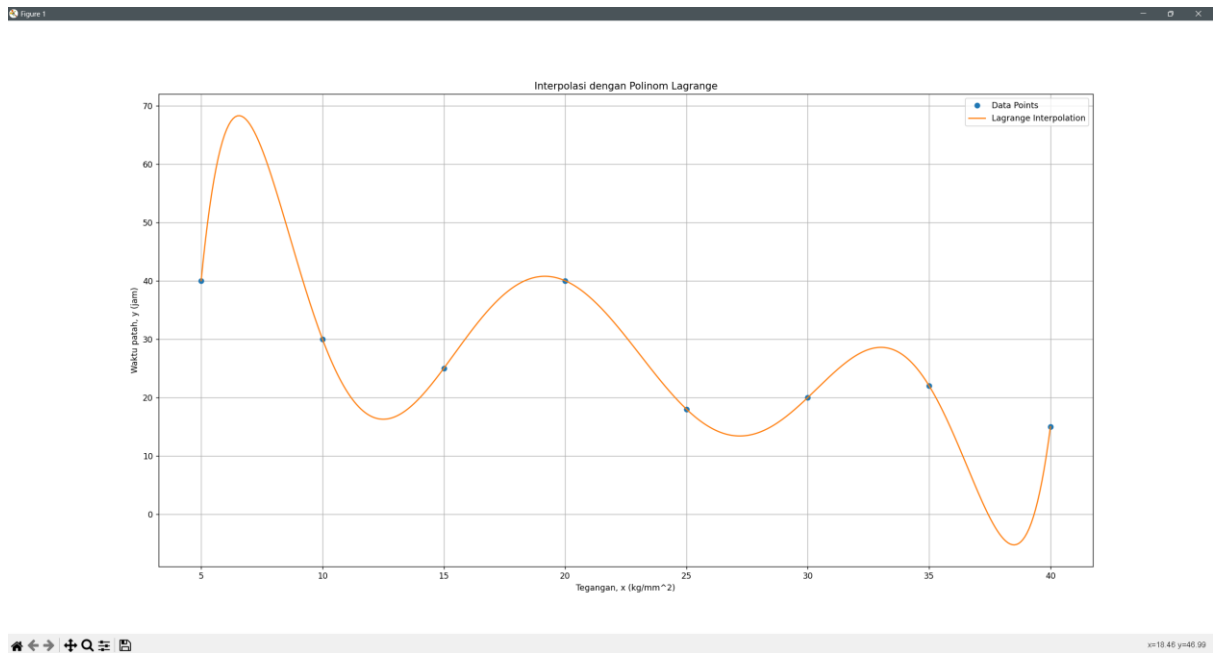
plt.show()

```

Keterangan:

- 'x' dan 'y' adalah data tegangan dan waktu patah
- Fungsi 'lagrange_interpolation' berfungsi untuk menghitung nilai interpolasi menggunakan polinom lagrange

Output:



2. Interpolasi Polinom Newton

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

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

# Fungsi untuk interpolasi Newton
def newton_interpolation(x_values, y_values, x):
    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, :]

coef = divided_diff(x_values, y_values)
n = len(x_values) - 1
p = coef[n]

for k in range(1, n + 1):
    p = coef[n - k] + (x - x_values[n - k]) * p

return p

# Plotting hasil interpolasi
x_range = np.linspace(5, 40, 400)
y_newton = [newton_interpolation(x, y, xi) for xi in
x_range]

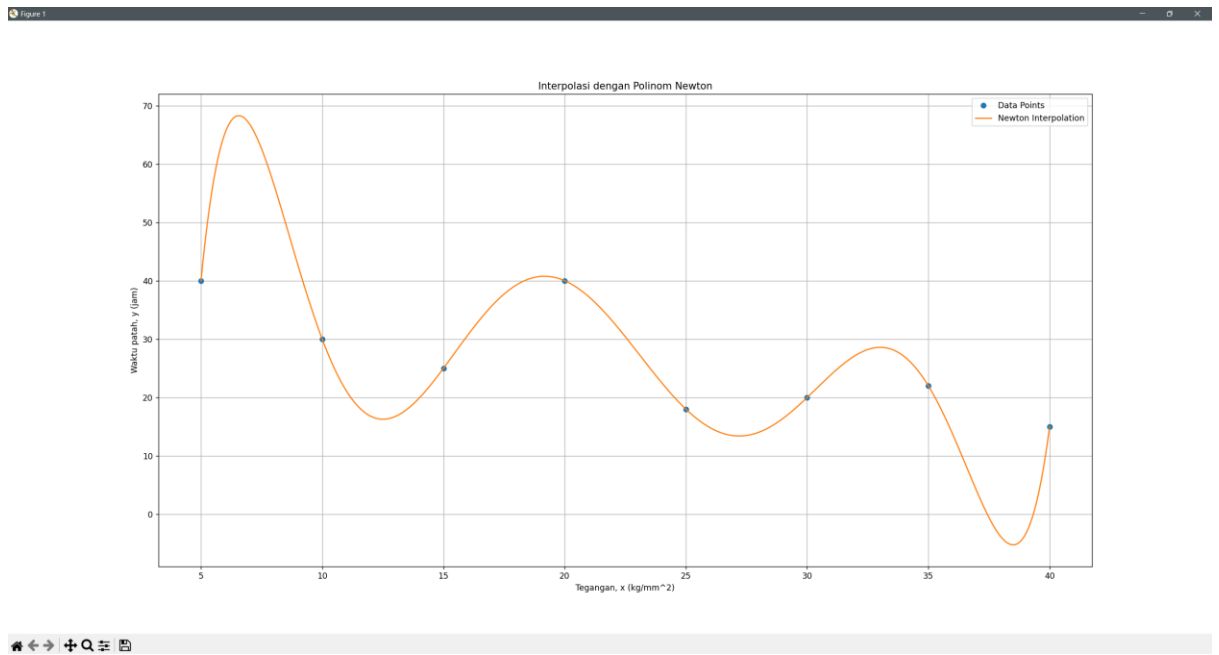
plt.figure(figsize=(12, 6))
plt.plot(x, y, 'o', label='Data Asli')
plt.plot(x_range, y_newton, label='Interpolasi Newton')
plt.xlabel('Tegangan, x (kg/mm^2)')
plt.ylabel('Waktu Patah, y (jam)')
plt.legend()
plt.title('Interpolasi dengan Polinom Newton')
plt.grid(True)
plt.show()

```

Keterangan:

- ‘x’ dan ‘y’ adalah data tegangan dan waktu patah
- Fungsi ‘newton_interpolation’ berfungsi untuk menghitung nilai interpolasi menggunakan polinom Newton

Output:



Dari hasil interpolasi di atas menunjukkan bahwa kedua metode memberikan hasil yang sama.