

TUGAS METODE NUMERIK
IMPLEMENTASI INTERPOLASI
POLINOM LAGRANGE

ZEKA EMO
21120122130075
2024/2025

Link GitHub: https://github.com/zekaemo/Implementasi-Polinomial_Metode-Numerik

Sebuah pengukuran fisika telah dilakukan untuk menentukan hubungan antara tegangan yang diberikan kepada baja tahan-karat dan waktu yang diperlukan hingga baja tersebut patah. delapa 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

I. Source Code

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

# Model interpolasi Lagrange
def lagrange_interpolation(f: list, xi: int, n: int) -> float:
    result = 0.0
    for i in range(n):
        term = f[i].y
```

```

        for j in range(n):
            if j != i:
                term = term * (xi - f[j].x) / (f[i].x - f[j].x)
            result += term
        return result

# Memasukkan Data
class Data:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __repr__(self):
        return f"Data({self.x}, {self.y})"
x = [5, 10, 15, 20, 25, 30, 35, 40]
y = [40, 30, 25, 40, 18, 20, 22, 15]
f = [Data(xi, yi) for xi, yi in zip(x, y)]

# Membuat garis interpolasi
x_garis = np.linspace(5, 40, 100)
y_garis = [lagrange_interpolation(f, point, 8) for point in x_garis]

plt.scatter(x, y)
plt.plot(x_garis, y_garis)

# Deployment
x_dep = np.array ([8, 13, 18, 23, 28, 33, 38])
y_dep = [lagrange_interpolation(f, x_dep, 8)]
y_dep

# visualisasi data
plt.figure(figsize=(10, 6))
plt.scatter (x,y,color="green", label="Titik yang diketahui")

```

```
plt.scatter(x_dep,y_dep, color="purple", label="Titik Uji",
zorder=5)
plt.plot (x_garis, y_garis, label= 'Polinomial Lagrange')
plt.title('Interpolasi Polinomial Newton')
plt.xlabel('Tegangan, x (kg/mm2)')
plt.ylabel('Waktu patah, y (jam)')
plt.grid(True)
```

II. Langkah-Langkah Pengerjaan

1. Import library

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

- Library Numpy digunakan untuk menjalankan operasi-operasi numerik.
- Library matplotlib.pyplot digunakan untuk visualisasi data dalam bentuk grafik.

2. Deklarasi fungsi model **Interpolasi Lagrange**

```
# Model interpolasi Lagrange
def lagrange_interpolation(f: list, xi: int, n: int)
-> float:
    result = 0.0
    for i in range(n):
        term = f[i].y
        for j in range(n):
            if j != i:
                term = term * (xi - f[j].x) / (f[i].x
- f[j].x)
        result += term
    return result
```

- Mendeklarasi fungsi lagrange_interpolation dengan parameter:

1. `f`: data latih dalam bentuk array
 2. `xi`: titik interpolasi
 3. `n`: jumlah titik interpolasi
- Inisialisasi hasil fungsi pada variabel `result` dengan nilai 0,0.
 - Menghitung setiap titik pada data latih (`f`) terhadap titik interpolasi yang ditentukan (`xi`) dan disimpan di variabel `term`.
 - Menghitung nilai interpolasi dengan mengalikan nilai `term` dengan rumus **Interpolasi Lagrange**. Setelahnya ini akan disimpan pada variabel `result` yang telah diinisialisasikan di awal.
 - Step ini akan dilakukan secara berulang sebanyak `n` kali.

3. Memasukkan data

```
class Data:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __repr__(self):
        return f"Data({self.x}, {self.y})"

x = [5, 10, 15, 20, 25, 30, 35, 40]
y = [40, 30, 25, 40, 18, 20, 22, 15]
f = [Data(xi, yi) for xi, yi in zip(x, y)]
```

- Mendeklarasikan kelas data dengan dua atribut: `x` dan `y` serta dua fungsi: `__init__` dan `__repr__`. Di mana fungsi `__repr__` digunakan untuk merepresentasikan string dari objek `Data`.

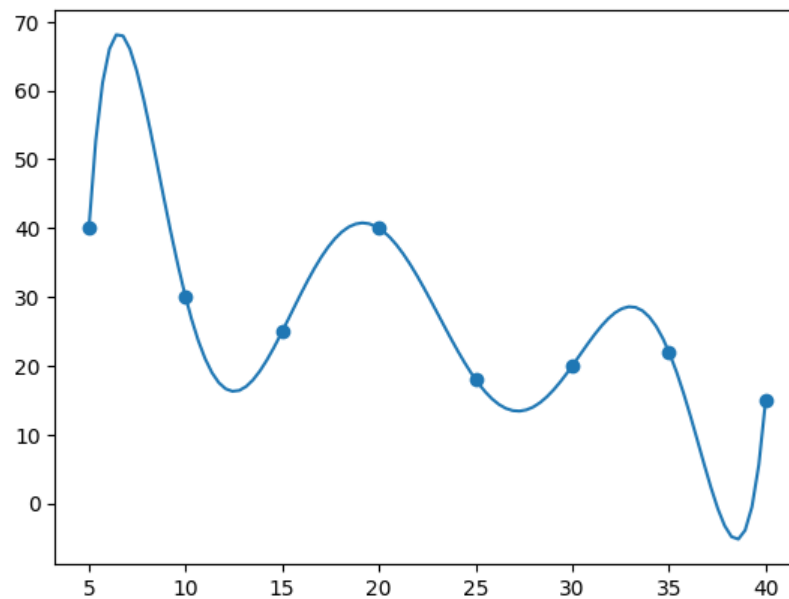
- Inisialisasi array x dan y yang merupakan data tegangan dan waktu patah.
- Membuat daftar objek f yang berisi objek-objek `Data` dengan menggunakan list comprehension. Setiap objek `Data` dibuat dengan menggunakan nilai x_i dan y_i dari setiap pasangan nilai dalam x dan y yang sesuai menggunakan fungsi `zip(x, y)`.
- Hasil dari daftar objek f , adalah sebagai berikut:
`[Data(5, 40), Data(10, 30), Data(15, 25), Data(20, 40), Data(25, 18), Data(30, 20), Data(35, 22), Data(40, 15)]`

4. Membuat garis interpolasi

```
x_garis = np.linspace(5, 40, 100)
y_garis = [lagrange_interpolation(f, point, 8) for
point in x_garis]

plt.scatter(x, y)
plt.plot(x_garis, y_garis)
```

- Variabel x_garis berisi nilai dari tiap titik pada data yang akan terpolasi. Dengan instruksi `np.linspace`, titik-titik akan terdistribusi merata dari 5 hingga 40 sebanyak 100 titik.
- Menghitung nilai y_garis interpolasi untuk setiap titik pada x_garis yang sudah di-interpolasi menggunakan fungsi `lagrange_interpolation`. Perhitungan ini menggunakan daftar f yang berisi objek-objek `Data` dan titik yang sedang diproses.
- Visualisasi hasil dari langkah keempat ini adalah sebagai berikut:



5. Pengujian dengan data baru (deployment)

```
x_dep = np.array ([8, 13, 18, 23, 28, 33, 38])
y_dep = [lagrange_interpolation(f,x_dep, 8)]
y_dep
```

- Melakukan interpolasi pada data asing dalam variabel `x_dep`.
- Hasil dari interpolasi `x_dep` akan disimpan pada variabel `y_dep`.
 - Dengan nilai `x_dep`: [8, 13, 18, 23, 28, 33, 38], hasil interpolasi yang akan tersimpan pada `y_dep`: [56.58121984, 16.72433664, 39.25446144, 27.77799424, 14.03733504, 28.59088384, -3.90695936]

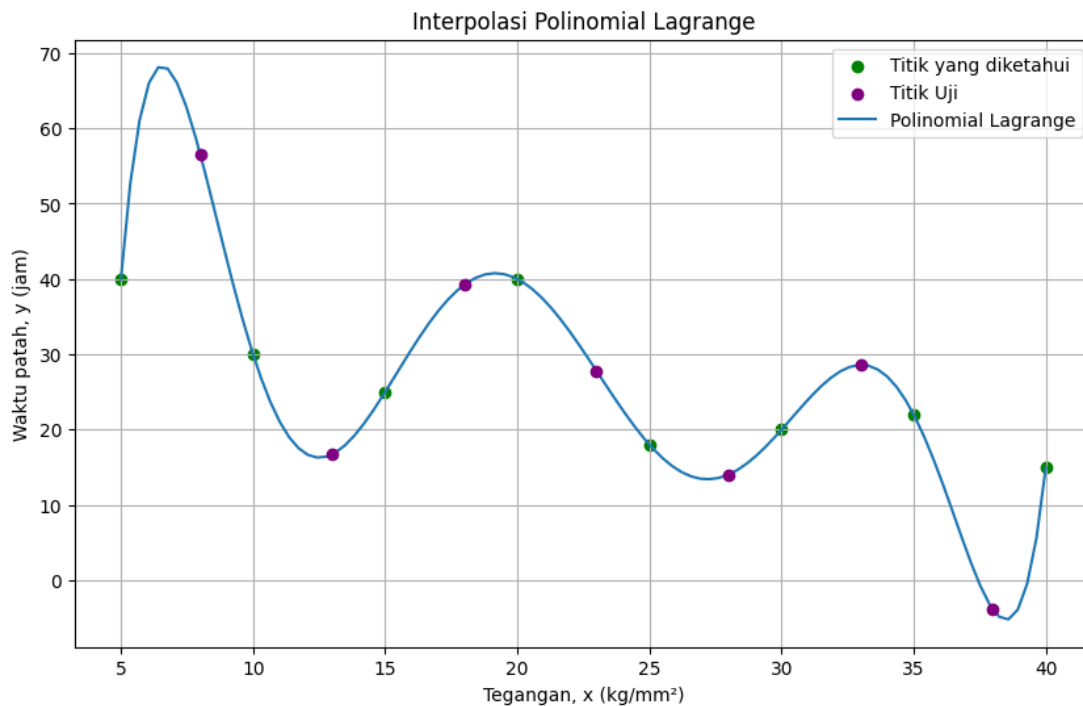
6. Visualisasi data

```
plt.figure(figsize=(10, 6))
plt.scatter (x,y,color="green", label="Titik yang diketahui")
```

```
plt.scatter(x_dep,y_dep, color="purple",
label="Titik Uji", zorder=5)
plt.plot (x_garis, y_garis, label= 'Polinomial
Lagrange')
plt.title('Interpolasi Polinomial Newton')
plt.xlabel('Tegangan, x (kg/mm2)')
plt.ylabel('Waktu patah, y (jam)')
plt.grid(True)
```

- Menyebar titik (x,y) dan (x_dep, y_dep) dengan instruksi `plt.scatter`.
- Menampilkan garis interpolasi dari (x_garis, y_garis) dengan instruksi `plt.plot`.

III. Analisis Hasil



Dalam grafik **Interpolasi Polinomial Lagrange**, terdapat tiga komponen utama:

1. Titik hijau, merupakan representasi data tegangan (x) terhadap waktu patah (y) yang sudah dideklarasikan sejak awal program berjalan. Data-data yang sudah dideklarasikan dari awal ini bisa kita sebut sebagai dataset. Dataset ini berfungsi sebagai acuan bagi model dalam menganalisa pola pada sistem.
2. Garis biru, merupakan garis yang dibentuk dengan melakukan proses interpolasi terhadap dataset yang diberikan.
3. Titik ungu, merupakan representasi data uji atau data asing yang belum dikenal oleh sistem sebelumnya. Pada prosesnya, sistem hanya diberi pengertian terkait besar tegangan (x). Proses ini nantinya akan menghasilkan nilai waktu patah (y) yang sesuai dengan masukan x . Sistem akan mencari nilai y dari garis biru terhadap titik x yang diuji.