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/mm2) | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 |
| Waktu patah, y (jam) | 40 | 30 | 25 | 40 | 18 | 20 | 22 | 15 |

1. 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) |

1. Langkah-Langkah Pengerjaan
2. 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.

1. 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.

1. 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)] |

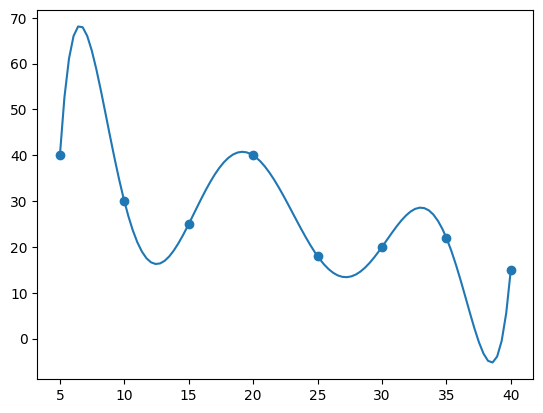
* Mendeklarasikan kelas data dengan dua atribut: x dan y serta dua fungsi: \_\_init\_\_ dan \_\_repr\_\_. Di mana fungsi \_\_repr\_\_ digunakan untuk merepresentasikan string dari objek Data.
* Inisialisasi array x dan y yang merupakan data tegangan dan waktu patah.
* Membuat daftar objek f yang berisi objek-objek Data dengan menggunakan list comprehension. Setiap objek Data dibuat dengan menggunakan nilai xi dan yi 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)]

1. 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:



1. 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 |

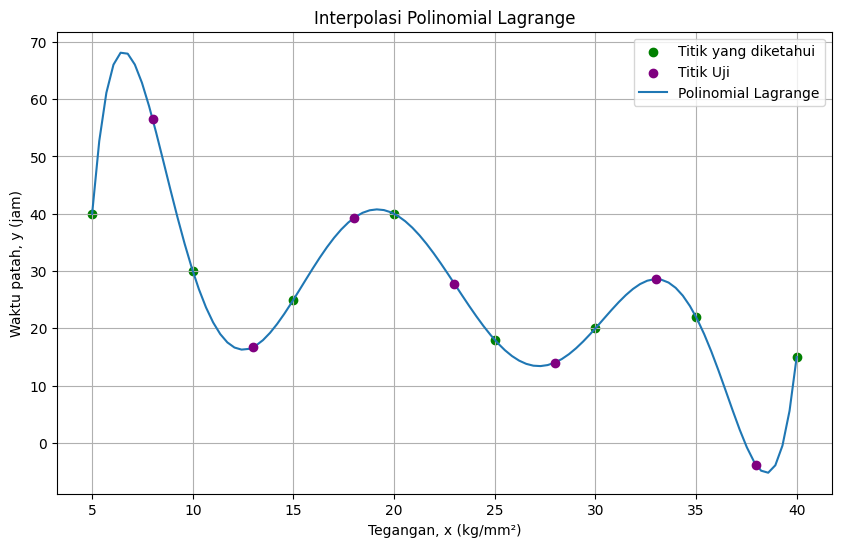
* Melakukan interpolasi pada data asing dalam variabel x\_dep.
* Hasil dari interpolasi x\_dep akan disimpan pada variabel y\_dep.
* Dengan nilai x\_dep: [8, 13, 18, 23, 28, 33, 38], hasil interpolasi yang akan tersimpan pada y\_dep: [56.58121984, 16.72433664, 39.25446144, 27.77799424, 14.03733504, 28.59088384,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

1. 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.

1. 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.