TUGAS METODE NUMERIK

Prabaswara Nasywa Maharani

21120122130076

Interpolasi Lagrange

**SOAL**

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 |

**SOURCE CODE**

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  # Fungsi Interpolasi Lagrange  def interpolasi\_lagrange(x, y, xi):  """  Menghitung polinomial interpolasi Lagrange pada titik xi.    :param x: daftar titik data x  :param y: daftar titik data y  :param xi: daftar atau skalar dari titik x untuk dievaluasi pada interpolasi  :return: nilai interpolasi pada xi  """  n = len(x)  yi = 0    for i in range(n):  # Menghitung polinomial basis L\_i(xi)  Li = 1  for j in range(n):  if i != j:  Li \*= (xi - x[j]) / (x[i] - x[j])  # Tambahkan nilai L\_i ke hasil interpolasi  yi += Li \* y[i]    return yi  # Fungsi untuk menguji interpolasi Lagrange  def uji\_interpolasi\_lagrange():  # Titik data dari tabel  x\_data = [5, 10, 15, 20, 25, 30, 35, 40]  y\_data = [40, 30, 25, 40, 18, 20, 22, 15]  # Rentang nilai x untuk interpolasi  x\_interpolated = np.linspace(5, 40, 500)    # Hitung nilai interpolasi y  y\_interpolated = [interpolasi\_lagrange(x\_data, y\_data, xi) for xi in x\_interpolated]    # Plot hasil interpolasi dan data asli  plt.figure(figsize=(10, 6))  plt.plot(x\_interpolated, y\_interpolated, label='Kurva Interpolasi Lagrange')  plt.scatter(x\_data, y\_data, color='red', label='Titik Data Asli')  plt.title('Interpolasi Lagrange: Hubungan Tegangan vs Waktu Patah')  plt.xlabel('Tegangan, x (kg/mm²)')  plt.ylabel('Waktu Patah, y (jam)')  plt.legend()  plt.grid(True)  plt.show()  data\_sets\_baru = [  ([1, 2, 3, 4, 5], [1, 4, 9, 16, 25], 'Fungsi Kuadrat y = x^2'),  ([1, 2, 3, 4, 5], [1, 8, 27, 64, 125], 'Fungsi Kubik y = x^3'),  ([0, 0.5, 1, 1.5, 2], [0, 0.4794, 0.8415, 0.9975, 0.9093], 'Fungsi Sinus y = sin(x)')  ]  # Menjalankan fungsi pengujian untuk verifikasi  uji\_interpolasi\_lagrange() |

**PENJELASAN SOURCE CODE & LANGKAH-LANGKAH**

1. Mengimpor modul:

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt |

* numpy: Digunakan untuk operasi numerik dan manipulasi data (array, matriks, fungsi matematika).
* matplotlib.pyplot: Digunakan untuk visualisasi data (plot, grafik).

1. Definisi Fungsi Interpolasi Lagrange

|  |
| --- |
| def interpolasi\_lagrange(x, y, xi):  """  Menghitung polinomial interpolasi Lagrange pada titik xi.    :param x: daftar titik data x  :param y: daftar titik data y  :param xi: nilai x untuk dievaluasi pada interpolasi  :return: nilai interpolasi pada xi  """  n = len(x) # Jumlah titik data  yi = 0 # Inisialisasi hasil interpolasi    for i in range(n):  # Inisialisasi polinomial basis L\_i(xi)  Li = 1  for j in range(n):  if i != j:  # Menghitung polinomial basis L\_i(xi)  Li \*= (xi - x[j]) / (x[i] - x[j])  # Menambahkan hasil L\_i(xi) \* y[i] ke hasil akhir  yi += Li \* y[i]    return yi |

1. Inisialisasi:

* n = len(x): Menghitung jumlah titik data.
* yi = 0: Menginisialisasi hasil interpolasi sebagai 0.

1. Menghitung Polinomial Basis:

* Loop for i in range(n): Iterasi melalui setiap titik data *x[i].*
* Li = 1: Inisialisasi polinomial basis *Li​*.
* Loop for j in range(n): Iterasi melalui setiap titik data lainnya *x[j].*
  + if i != j: Pastikan *j* ≠ *i* untuk menghindari pembagian dengan nol.
  + Li \*= (xi - x[j]) / (x[i] - x[j]): Menghitung produk rasio untuk polinomial basis LiL\_iLi​.

1. Mengakumulasi Hasil:

* yi += Li \* y[i]: Mengalikan polinomial basis *Li* dengan *y[i]* dan menambahkannya ke hasil interpolasi yi.

1. Mengembalikan Nilai:

* return yi: Mengembalikan nilai interpolasi pada xi

1. Menguji dan memvisualisasikan interpolasi Lagrange dengan data:

|  |
| --- |
| def uji\_interpolasi\_lagrange():  # Titik data dari tabel  x\_data = [5, 10, 15, 20, 25, 30, 35, 40]  y\_data = [40, 30, 25, 40, 18, 20, 22, 15]  # Rentang nilai x untuk interpolasi  x\_interpolated = np.linspace(5, 40, 500)    # Titik uji untuk evaluasi  x\_test = [8, 18, 28, 38]  y\_test = [interpolasi\_lagrange(x\_data, y\_data, xi) for xi in x\_test]    # Hitung nilai interpolasi y  y\_interpolated = [interpolasi\_lagrange(x\_data, y\_data, xi) for xi in x\_interpolated]    # Plot hasil interpolasi dan data asli  plt.figure(figsize=(10, 6))  plt.plot(x\_interpolated, y\_interpolated, label='Kurva Interpolasi Lagrange')  plt.scatter(x\_data, y\_data, color='red', label='Titik Data Asli')  plt.scatter(x\_test, y\_test, color='blue', marker='x', label='Titik Uji Interpolasi')  plt.title('Interpolasi Lagrange: Hubungan Tegangan vs Waktu Patah')  plt.xlabel('Tegangan, x (kg/mm²)')  plt.ylabel('Waktu Patah, y (jam)')  plt.legend()  plt.grid(True)  plt.show()  # Menjalankan fungsi pengujian untuk verifikasi  uji\_interpolasi\_lagrange() |

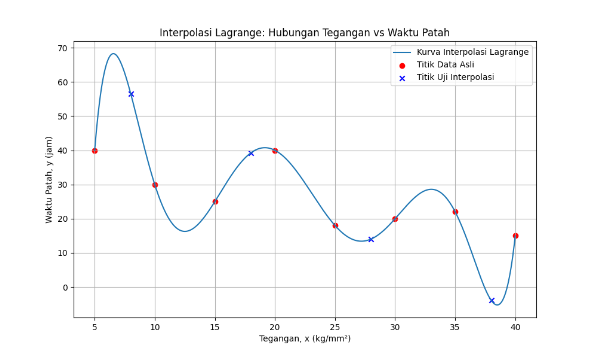
* 1. Data Input:
* x\_data dan y\_data: Titik data asli pada sumbu xxx dan yyy.
  1. Menyiapkan Rentang untuk Interpolasi:
* x\_interpolated = np.linspace(5, 40, 500): Membuat 500 nilai xxx yang tersebar merata antara 5 dan 40 untuk menggambar kurva interpolasi.
  1. Menghitung Nilai Interpolasi pada Titik Uji:
* x\_test = [8, 18, 28, 38]: Menetapkan beberapa titik uji xxx yang spesifik.
* y\_test = [interpolasi\_lagrange(x\_data, y\_data, xi) for xi in x\_test]: Menghitung nilai interpolasi yyy pada titik-titik uji ini menggunakan fungsi interpolasi\_lagrange.
  1. Menghitung Kurva Interpolasi:
* y\_interpolated = [interpolasi\_lagrange(x\_data, y\_data, xi) for xi in x\_interpolated]: Menghitung nilai interpolasi yyy untuk rentang nilai xxx yang telah dibuat.
  1. Membuat Grafik:
* plt.figure(figsize=(10, 6)): Membuat gambar baru dengan ukuran 10x6 inci.
* plt.plot(x\_interpolated, y\_interpolated, label='Kurva Interpolasi Lagrange'): Menggambar kurva interpolasi.
* plt.scatter(x\_data, y\_data, color='red', label='Titik Data Asli'): Menampilkan titik data asli sebagai titik merah.
* plt.scatter(x\_test, y\_test, color='blue', marker='x', label='Titik Uji Interpolasi'): Menampilkan titik uji interpolasi sebagai tanda silang biru.
* plt.title('Interpolasi Lagrange: Hubungan Tegangan vs Waktu Patah'): Menetapkan judul grafik.
* plt.xlabel('Tegangan, x (kg/mm²)'): Menetapkan label sumbu x.
* plt.ylabel('Waktu Patah, y (jam)'): Menetapkan label sumbu y.
* plt.legend(): Menampilkan legenda.
* plt.grid(True): Menambahkan grid pada grafik.
* plt.show(): Menampilkan grafik.

1. Menjalankan fungsi pengujian :

|  |
| --- |
| uji\_interpolasi\_lagrange() |

Memanggil fungsi uji\_interpolasi\_lagrange untuk memverifikasi hasil interpolasi dan menampilkan grafik.

**ANALISIS HASIL GRAFIK**



* Kurva Interpolasi Lagrange

Garis yang dihasilkan oleh interpolasi Lagrange yang melewati semua titik data asli. Ini menunjukkan bagaimana polinomial interpolasi sesuai dengan data.

* Titik Data Asli

Titik merah di grafik adalah data asli. Mereka menunjukkan nilai sebenarnya yang diberikan atau diukur.

* Titik Uji Interpolasi

Tanda silang biru menunjukkan titik di mana interpolasi dihitung untuk memverifikasi interpolasi di antara titik-titik data asli.

Grafik di atas menampilkan hasil dari interpolasi Lagrange yang digunakan untuk memodelkan hubungan antara tegangan dan waktu patah. Garis melengkung biru adalah hasil dari interpolasi, untuk menyesuaikan semua titik data yang diberikan. Tujuannya adalah untuk memberikan gambaran keseluruhan tentang bagaimana tegangan memengaruhi waktu patah. Titik-titik merah menunjukkan titik-titik data asli yang diamati atau diukur, memberikan gambaran tentang distribusi sebenarnya dari hubungan tersebut. Titik-titik biru dengan tanda silang adalah titik-titik yang diuji untuk memeriksa seberapa baik garis interpolasi merepresentasikan hubungan tersebut di luar titik-titik data yang diberikan.