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 matplotlib.pyplot as plt  import numpy as np  # Data yang diberikan  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, y):  """ Menghitung tabel divided difference """  n = len(y)  coef = np.zeros([n, n])  coef[:,0] = y  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[i + j] - x[i])    return coef[0, :]  def newton\_poly(coef, x\_data, x):  """ Mengevaluasi polinomial Newton pada titik x yang diberikan """  n = len(x\_data) - 1  p = coef[n]  for k in range(1, n + 1):  p = coef[n - k] + (x - x\_data[n - k]) \* p  return p  # Menghitung koefisien divided difference  coefficients = divided\_diff(x, y)  # Mendefinisikan rentang nilai x untuk plotting interpolasi  x\_range = np.linspace(min(x), max(x), 100)  y\_range = [newton\_poly(coefficients, x, xi) for xi in x\_range]  # Plotting data asli dan polinomial interpolasi  plt.figure(figsize=(10, 6))  plt.plot(x, y, 'ro', label='Data points')  plt.plot(x\_range, y\_range, 'b-', label='Newton Interpolated Polynomial')  # Menguji interpolasi dengan beberapa nilai baru  x\_test = np.array([7, 12, 28])  y\_test = [newton\_poly(coefficients, x, xi) for xi in x\_test]  # Plotting titik uji  plt.plot(x\_test, y\_test, 'go', label='Test points')  # Detail tambahan untuk plot  plt.xlabel('Tegangan, x (kg/mm²)')  plt.ylabel('Waktu patah, y (jam)')  plt.legend()  plt.title('Newton Interpolation of Tegangan vs Waktu patah')  plt.grid(True)  plt.show() |

**Langkah-langkah**

1. Mengimpor modul yang diperlukan:

|  |
| --- |
| 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 Newton

|  |
| --- |
| def divided\_diff(x, y):  """ Menghitung tabel divided difference """  n = len(y)  coef = np.zeros([n, n])  coef[:,0] = y  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[i + j] - x[i])    return coef[0, :] |

* + Membentuk matriks coef yang diisi dengan nilai-nilai y pada kolom pertama.
  + Untuk setiap kolom berikutnya, nilai dihitung berdasarkan perbedaan yang dibagi dari kolom sebelumnya.

1. Mengevaluasi polinomial Newton pada titik x yang diberikan.:

|  |
| --- |
| def newton\_poly(coef, x\_data, x):  """ Mengevaluasi polinomial Newton pada titik x yang diberikan """  n = len(x\_data) - 1  p = coef[n]  for k in range(1, n + 1):  p = coef[n - k] + (x - x\_data[n - k]) \* p  return p |

* Polinomial Newton dievaluasi menggunakan koefisien yang diperoleh dari tabel divided differences.
* Metode ini menambahkan koefisien satu per satu mulai dari yang tertinggi, dikalikan dengan (x - x\_data) secara rekursif.

1. Menghitung Koefisien dan Rentang Plot:

|  |
| --- |
| coefficients = divided\_diff(x, y)  x\_range = np.linspace(min(x), max(x), 100)  y\_range = [newton\_poly(coefficients, x, xi) for xi in x\_range] |

* coefficients dihitung menggunakan divided\_diff dari data x dan y.
* x\_range adalah rentang nilai x untuk visualisasi interpolasi.
* y\_range dihitung dengan mengevaluasi polinomial Newton pada setiap xi di x\_range.

1. Membuat Plot:

|  |
| --- |
| # Plotting data asli dan polinomial interpolasi  plt.figure(figsize=(10, 6))  plt.plot(x, y, 'ro', label='Data points')  plt.plot(x\_range, y\_range, 'b-', label='Newton Interpolated Polynomial')  # Plotting titik uji  plt.plot(x\_test, y\_test, 'go', label='Test points') |

* figsize=(10, 6): Menentukan ukuran grafik menjadi 10 x 6 inci.
* 'ro': Menggambar titik data asli sebagai titik merah (r untuk warna merah, o untuk bentuk titik).
* 'b-': Menggambar kurva interpolasi Newton sebagai garis biru (b untuk warna biru, - untuk garis lurus).

1. Menambahkan Garis Interpolasi:

|  |
| --- |
| # Menguji interpolasi dengan beberapa nilai baru  x\_test = np.array([7, 12, 28])  y\_test = [newton\_poly(coefficients, x, xi) for xi in x\_test]  # Plotting titik uji  plt.plot(x\_test, y\_test, 'go', label='Test points') |

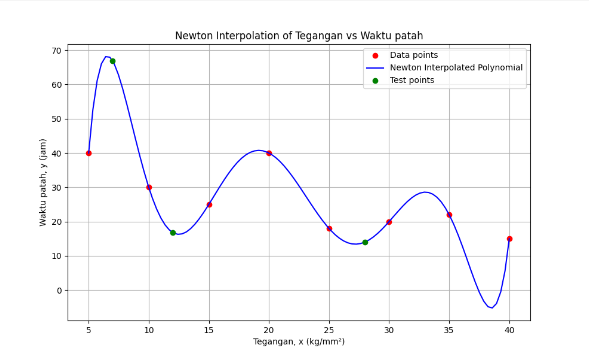
* x\_test: Array yang berisi nilai x untuk pengujian interpolasi, yaitu [7, 12, 28].
* y\_test: Nilai interpolasi y yang dihitung untuk setiap nilai x\_test menggunakan fungsi newton\_poly.
* 'go': Menggambar titik uji sebagai titik hijau (g untuk warna hijau, o untuk bentuk titik).

1. Pengujian dengan Data Baru:

|  |
| --- |
| # Detail tambahan untuk plot  plt.xlabel('Tegangan, x (kg/mm²)')  plt.ylabel('Waktu patah, y (jam)')  plt.legend()  plt.title('Newton Interpolation of Tegangan vs Waktu patah')  plt.grid(True)  plt.show() |

* xlabel: Menambahkan label pada sumbu X (Tegangan, x (kg/mm²)).
* ylabel: Menambahkan label pada sumbu Y (Waktu patah, y (jam)).
* legend(): Menambahkan legenda untuk menjelaskan elemen grafik (data asli, polinomial interpolasi, titik uji).
* title(): Menambahkan judul pada grafik (Newton Interpolation of Tegangan vs Waktu patah).
* grid(True): Mengaktifkan grid pada grafik untuk memudahkan pembacaan data.

**Analisis Hasil**



1. Data Asli (Titik Merah):

Data asli ditampilkan sebagai titik merah pada grafik. Titik-titik ini menunjukkan pasangan nilai Tegangan dan Waktu patah yang digunakan sebagai input untuk model interpolasi.

1. Kurva Polinomial Interpolasi Newton (Garis Biru):

Garis biru pada grafik mewakili polinomial interpolasi Newton yang dihasilkan. Kurva ini melewati semua titik data asli, sesuai dengan sifat interpolasi Newton yang mencoba meminimalkan deviasi dari data asli.

1. Titik Uji (Titik Hijau):

Titik uji (titik hijau) digunakan untuk memvalidasi interpolasi. Hasil interpolasi untuk nilai baru ini tampaknya sesuai dengan pola yang ditunjukkan oleh data asli dan kurva interpolasi, menunjukkan bahwa interpolasi Newton memberikan hasil yang wajar untuk nilai uji dalam rentang data asli.