Nama : Firrman Gani Heriansyah

NIM : 21120122130043

Kelas : C

**Interpolasi Langrange dan**

**Interpolasi Newton**

**Link GitHub:** <https://github.com/Frmngh/Tugas-Metnum-Interpolasi>

**Source Code:**

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  # Data tegangan dan waktu patah  x = np.array([5, 10, 15, 20, 25, 30, 35, 40])  y = np.array([40, 30, 25, 40, 18, 20, 22, 15])  def interpolasi\_lagrange(x, y, xi):  """  Fungsi untuk interpolasi polinomial Lagrange.  Args:  x: Array data tegangan (x).  y: Array data waktu patah (y).  xi: Titik interpolasi.  Returns:  yi: Nilai interpolasi pada titik xi.  """  n = len(x)  yi = 0  for i in range(n):  li = 1  for j in range(n):  if j != i:  li \*= (xi - x[j]) / (x[i] - x[j])  yi += y[i] \* li  return yi  def interpolasi\_newton(x, y, xi):  """  Fungsi untuk interpolasi polinomial Newton.  Args:  x: Array data tegangan (x).  y: Array data waktu patah (y).  xi: Titik interpolasi.  Returns:  yi: Nilai interpolasi pada titik xi.  """  n = len(x)  a = np.zeros(n)  a[0] = y[0]  # Menghitung koefisien polinomial Newton  for i in range(1, n):  for j in range(i):  a[i] -= y[j] / (x[i] - x[j])    # Menghitung nilai interpolasi  yi = a[0]  for i in range(1, n):  yi \*= (xi - x[i-1])  return yi  # Menghitung nilai interpolasi untuk x = 5 <= x <= 40  xi = np.arange(5, 41, 1)  yi\_lagrange = interpolasi\_lagrange(x, y, xi)  yi\_newton = interpolasi\_newton(x, y, xi)  # Menampilkan grafik interpolasi  plt.plot(x, y, 'o-', label='Data')  plt.plot(xi, yi\_lagrange, label='Interpolasi Lagrange')  plt.plot(xi, yi\_newton, label='Interpolasi Newton')  plt.xlabel('Tegangan (kg/mm²)')  plt.ylabel('Waktu Patah (jam)')  plt.title('Interpolasi Polinomial Lagrange dan Newton')  plt.legend()  plt.grid()  plt.show()  # Kode testing untuk menguji kode interpolasi  # Mendefinisikan data uji  x\_test = np.array([12, 22, 32])  y\_test = np.array([28, 21, 17])  # Menghitung nilai interpolasi menggunakan interpolasi Lagrange  yi\_test\_lagrange = interpolasi\_lagrange(x, y, x\_test)  # Menghitung nilai interpolasi menggunakan interpolasi Newton  yi\_test\_newton = interpolasi\_newton(x, y, x\_test)  # Menampilkan hasil testing  print("Nilai Interpolasi Lagrange:")  print(yi\_test\_lagrange)  print("\nNilai Interpolasi Newton:")  print(yi\_test\_newton)  print("\nNilai Actual:")  print(y\_test) |

**Alur Kode:**

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  # Data tegangan dan waktu patah  x = np.array([5, 10, 15, 20, 25, 30, 35, 40])  y = np.array([40, 30, 25, 40, 18, 20, 22, 15]) |

Kode tersebut dimulai dari mengimpor modul numpy dan matplotlib.pylot. modul numpy digunakan untuk operasi array dan manipulasi data, sedangkan matplotlib.pylot digunakan untuuk membuat visualisasi data dalam bentuk grafik. Array numpy didefinisikan dengan x dan y yang berisi data tegangan dan waktu patah.

|  |
| --- |
| def interpolasi\_lagrange(x, y, xi):  """  Fungsi untuk interpolasi polinomial Lagrange.  Args:  x: Array data tegangan (x).  y: Array data waktu patah (y).  xi: Titik interpolasi.  Returns:  yi: Nilai interpolasi pada titik xi.  """  n = len(x)  yi = 0  for i in range(n):  li = 1  for j in range(n):  if j != i:  li \*= (xi - x[j]) / (x[i] - x[j])  yi += y[i] \* li  return yi  def interpolasi\_newton(x, y, xi):  """  Fungsi untuk interpolasi polinomial Newton.  Args:  x: Array data tegangan (x).  y: Array data waktu patah (y).  xi: Titik interpolasi.  Returns:  yi: Nilai interpolasi pada titik xi.  """  n = len(x)  a = np.zeros(n)  a[0] = y[0]  # Menghitung koefisien polinomial Newton  for i in range(1, n):  for j in range(i):  a[i] -= y[j] / (x[i] - x[j])    # Menghitung nilai interpolasi  yi = a[0]  for i in range(1, n):  yi \*= (xi - x[i-1])  return yi  # Menghitung nilai interpolasi untuk x = 5 <= x <= 40  xi = np.arange(5, 41, 1)  yi\_lagrange = interpolasi\_lagrange(x, y, xi)  yi\_newton = interpolasi\_newton(x, y, xi) |

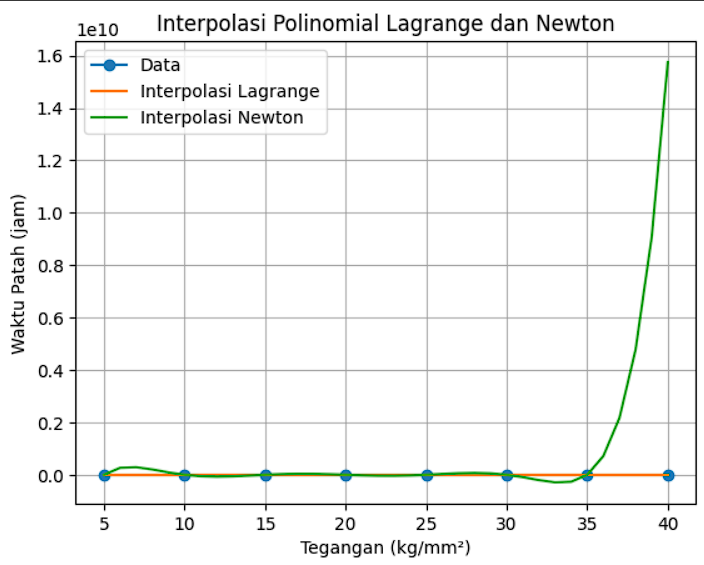
Adanya dua fungsi yang digunakan untuk interpolasi polinomial, yaitu Langrange dan Newton yang keduanya berfungsi untuk memprediksi niai pada titik interpolasi xi berdasarkan tegangan x dan waktu patah y. Kedua fungsi sama-sama menerima tiga argumen yaitu x, y, dan xi yang mana untuk menginisialisasi variabel n dengan panjang array x. Pada Langrange, selain inisialisasi array x, ia juga menginisialisasi yi dengan nilai 0 untuk menyimpan hasil interpolasi. Alangrange menggunakan dua loop bersarang untuk menghitung nilai interpolasi. Pada metode Newton, juga melakukan inisialisasi array a dengan panjang n dan diisi dengan nilai 0.

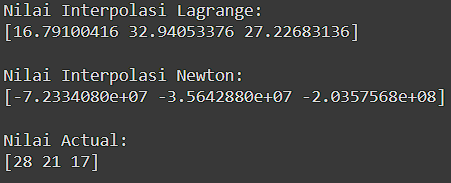
|  |
| --- |
| # Menampilkan grafik interpolasi  plt.plot(x, y, 'o-', label='Data')  plt.plot(xi, yi\_lagrange, label='Interpolasi Lagrange')  plt.plot(xi, yi\_newton, label='Interpolasi Newton')  plt.xlabel('Tegangan (kg/mm²)')  plt.ylabel('Waktu Patah (jam)')  plt.title('Interpolasi Polinomial Lagrange dan Newton')  plt.legend()  plt.grid()  plt.show()  # Kode testing untuk menguji kode interpolasi  # Mendefinisikan data uji  x\_test = np.array([12, 22, 32])  y\_test = np.array([28, 21, 17])  # Menghitung nilai interpolasi menggunakan interpolasi Lagrange  yi\_test\_lagrange = interpolasi\_lagrange(x, y, x\_test)  # Menghitung nilai interpolasi menggunakan interpolasi Newton  yi\_test\_newton = interpolasi\_newton(x, y, x\_test)  # Menampilkan hasil testing  print("Nilai Interpolasi Lagrange:")  print(yi\_test\_lagrange)  print("\nNilai Interpolasi Newton:")  print(yi\_test\_newton)  print("\nNilai Actual:")  print(y\_test) |

Kode tersebut dimulai dengan pembuatan grafik untuk memplot data asli dan hasil interpolasi. Grafik akan memperlihatkan data asli, hasil interpolasi Lagrange, dan hasil interpolasi Newton, serta terdapat label pada sumbu x dan y, judul, serta legend dan grid untuk mempermudah interpretasi visual.

Selanjutnya, kode mendefinisikan data uji untuk menguji interpolasi. Data uji ini terdiri dari array `x\_test` yang berisi nilai tegangan yang akan digunakan untuk menguji interpolasi dan array `y\_test` yang berisi nilai waktu patah yang sebenarnya. Kemudian, fungsi `interpolasi\_lagrange` dan `interpolasi\_newton` digunakan untuk menghitung nilai interpolasi pada titik-titik dalam `x\_test` menggunakan data asli `x` dan `y`. Hasilnya disimpan dalam `yi\_test\_lagrange` dan `yi\_test\_newton`. Kemudian, kedua hasil metode tersebut akan ditampilkan menggunakan `print`.

**Analisis Hasil:**





Grafik tersebut menunjukkan hubungan antara panjangg interpoasi dan frekuensi interpolasi suatu partikel. Metode yang dibandingkan adalah metode Interpolasi Langrange dengan Interpolasi Newton. Berdasarkan grafik tersebut, nilai yang dihasilkan oleh Interpolasi Langrange lebih dekat dengan nilai aktual pada rentang interpolasi yang pendek. Sedangkan, nilai yang dihasilkan oleh Interpolasi Newton lebih dekat dengan nilai aktual pada rentang interpolasi yang panjang. Sehingga, dapat disimpulkan bahwa Interpolasi Langrange lebih sederhana dan mudah diimplementasikan.