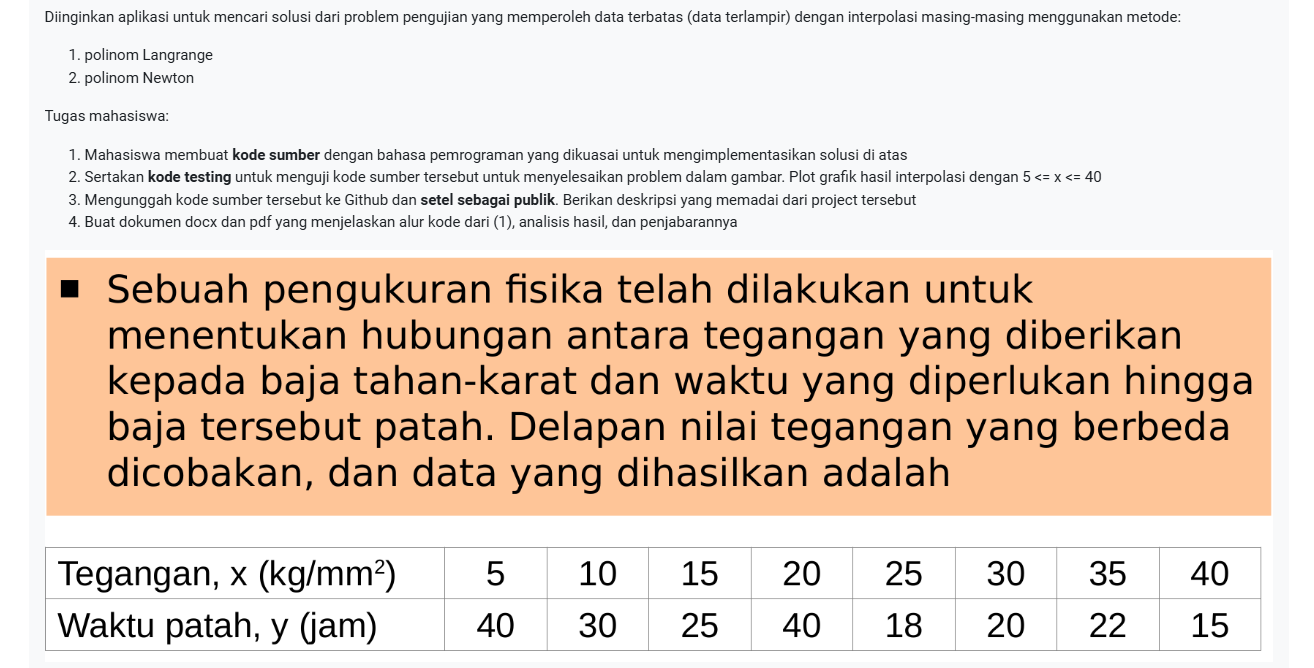
Nama : Finodya Yahdun

NIM : 21120122130065

Mata Kuliah : Metode Numerik

Kelas : D



|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  # Data dari gambar  x\_data = np.array([5, 10, 15, 20, 25, 30, 35, 40])  y\_data = np.array([40, 30, 25, 40, 18, 20, 22, 15])  # Interpolasi Polinom Lagrange  def interpolasi\_lagrange(x, x\_titik, y\_titik):      def L(k, x):          term = [(x - x\_titik[j]) / (x\_titik[k] - x\_titik[j]) for j in range(len(x\_titik)) if j != k]          return np.prod(term, axis=0)      return np.sum([y\_titik[k] \* L(k, x) for k in range(len(x\_titik))], axis=0)  # Interpolasi Polinom Newton  def interpolasi\_newton(x, x\_titik, y\_titik):      def beda\_terbagi(x\_titik, y\_titik):          n = len(y\_titik)          coef = np.zeros([n, n])          coef[:,0] = y\_titik          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\_titik[i+j] - x\_titik[i])          return coef[0,:]      koef = beda\_terbagi(x\_titik, y\_titik)      n = len(koef)      hasil = koef[0]      for i in range(1, n):          term = koef[i]          for j in range(i):              term \*= (x - x\_titik[j])          hasil += term      return hasil  # Menguji fungsi dan plot hasilnya  x\_range = np.linspace(5, 40, 100)  y\_lagrange = interpolasi\_lagrange(x\_range, x\_data, y\_data)  y\_newton = interpolasi\_newton(x\_range, x\_data, y\_data)  # Membuat grafik hasil interpolasi  plt.figure(figsize=(10, 6))  plt.plot(x\_data, y\_data, 'o', label='Titik Data')  plt.plot(x\_range, y\_lagrange, '-', label='Interpolasi Lagrange')  plt.plot(x\_range, y\_newton, '--', label='Interpolasi Newton')  plt.xlabel('Tegangan, x (kg/mm²)')  plt.ylabel('Waktu patah, y (jam)')  plt.title('Interpolasi Waktu Patah vs. Tegangan')  plt.legend()  plt.grid(True)  plt.show() |

Penjelasan Source Code

Source code diatas untuk mengimplementasikan interpolasi polinomial menggunakan metode Lagrange dan Newton, serta memvisualisasikan hasilnya. Pertama, library `numpy` dan `matplotlib.pyplot` diimpor untuk mendukung operasi numerik dan visualisasi data. `numpy` menyediakan fungsi-fungsi numerik yang efisien, sementara `matplotlib.pyplot` digunakan untuk membuat grafik. Data titik yang digunakan untuk interpolasi didefinisikan dalam bentuk array `x\_data` dan `y\_data`. Array ini berisi nilai-nilai yang akan digunakan sebagai titik referensi untuk membangun polinom interpolasi. Fungsi `interpolasi\_lagrange` digunakan untuk menghitung nilai interpolasi menggunakan metode Lagrange. Di dalamnya, fungsi basis Lagrange `L(k, x)` didefinisikan untuk menghitung produk dari selisih nilai x yang diperlukan untuk polinom. Hasil akhir interpolasi diperoleh dengan menjumlahkan hasil perkalian antara nilai y dan basis Lagrange untuk setiap titik data.

Fungsi `interpolasi\_newton` digunakan untuk menghitung nilai interpolasi menggunakan metode Newton. Dalam fungsi ini, tabel beda terbagi dihitung melalui fungsi `beda\_terbagi`, yang menghasilkan koefisien polinom Newton. Polinom Newton kemudian dibangun menggunakan koefisien ini dan hasilnya adalah nilai interpolasi pada titik-titik yang diberikan. Untuk menguji fungsi interpolasi, rentang nilai x dari 5 hingga 40 didefinisikan sebagai `x\_range`. Fungsi interpolasi Lagrange dan Newton diaplikasikan pada rentang nilai ini untuk menghasilkan nilai y yang diinterpolasi, yaitu `y\_lagrange` dan `y\_newton`. Akhirnya, hasil interpolasi dan data titik asli divisualisasikan dalam sebuah grafik. Grafik ini menampilkan titik data asli dengan simbol 'o', serta kurva interpolasi Lagrange dan Newton dengan garis yang berbeda. Label sumbu, judul grafik, dan legenda ditambahkan untuk memudahkan interpretasi, dan grid juga ditampilkan untuk membantu pembacaan grafik. Grafik ini kemudian ditampilkan menggunakan fungsi `plt.show()`. Source code tersebut mengimplementasikan dan membandingkan interpolasi polinomial menggunakan metode Lagrange dan Newton, serta memvisualisasikan hasilnya dalam sebuah grafik.

