

TUGAS METODE NUMERIK
IMPLEMENTASI INTERPOLASI
POLINOM NEWTON

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/mm ²) | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 |
| Waktu patah, y (jam) | 40 | 30 | 25 | 40 | 18 | 20 | 22 | 15 |

I. Source Code

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

def divided_diff(x, y):
    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

def newton_poly(coef, x_data, x):
    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

#memasukkan data
x = np.array([5, 10, 15, 20, 25, 30, 35, 40])
y = np.array([40, 30, 25, 40, 18, 20, 22, 15])

# cari nilai selisih terbagi
dd= divided_diff(x, y) [0, :]
dd

# membuat garis interpolasi
x_garis = np.linspace(5, 40, 100)
y_garis = [newton_poly(dd,x, point) 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= newton_poly (dd, x, x_test)

# visualisasi
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 Newton')
plt.title('Interpolasi Polinomial Newton')
plt.xlabel('Tegangan, x (kg/mm2)')
plt.ylabel('Waktu patah, y (jam)')
plt.grid(True)
```

II. Alur kode

1. 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.

2. Deklarasi fungsi model **Perbedaan Terbagi**

```
def divided_diff(x, y):
    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
```

- Deklarasi fungsi **Perbedaan Terbagi** dengan dua parameter: x dan y yang merupakan data tegangan (x) dan waktu patah (y).
- Inisialisasi variabel coef yang akan menyimpan perhitungan perbedaan terbagi.

- Perhitungan ini akan dilakukan berulang kali sebanyak data y yang diberikan.

3. Deklarasi fungsi model **Polinomial Newton Forward**

```
def newton_poly(coef, x_data, x):
    '''
    evaluate the newton polynomial
    at x
    '''
    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
```

- Deklarasi fungsi **Polinomial Newton Forward** dengan tiga parameter: coef yang merupakan nilai perbedaan terbagi, x_data yang merupakan data x pada dataset, dan x yang merupakan titik yang akan dihitung nilai interpolasinya.
- Inisialisasi variabel n dan p:
 1. n adalah derajat polinomial interpolasi, yang dihitung dari jumlah titik data dikurangi satu.
 2. p diinisialisasi dengan koefisien tertinggi dari polinomial, yaitu coef[n].
- Perhitungan nilai polinomial akan disimpan pada variabel p, dan berulang kali dengan iterasi mundur sebanyak derajat polinomial n.

4. Memasukkan data

```
x = np.array([5, 10, 15, 20, 25, 30, 35, 40])
y = np.array([40, 30, 25, 40, 18, 20, 22, 15])
```

- Inisialisasi array x dan y yang merupakan data tegangan dan waktu patah.

5. Mencari nilai perbedaan terbagi

```
dd= divided_diff(x, y) [0, :]
```

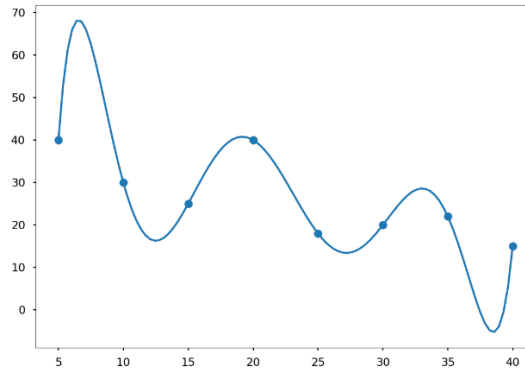
- Melakukan perhitungan perbedaan terbagi dengan fungsi `divided_diff` pada array `x` dan `y`.
- Menggunakan indeks `[0, :]` untuk memilih baris pertama dari tabel perbedaan terbagi.
- Nilai perbedaan terbagi dari `x` dan `y`:
[4.00000000e+01, -2.00000000e+00, 1.00000000e-01,
2.00000000e-02, -4.80000000e-03, 5.06666667e-04, -
3.49333333e-05, 1.76761905e-06]

6. Membuat garis interpolasi

```
x_garis = np.linspace(5, 40, 100)  
y_garis = [newton_poly(dd,x, point) 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 `newton_poly`.

- Visualisasi hasil dari langkah keempat ini adalah sebagai berikut:



7. Pengujian dengan data baru (deployment)

```
x_dep= np.array ([8, 13, 18, 23, 28, 33, 38])
y_dep= newton_poly (dd, x, x_test)
```

- 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]

8. Visualisasi data

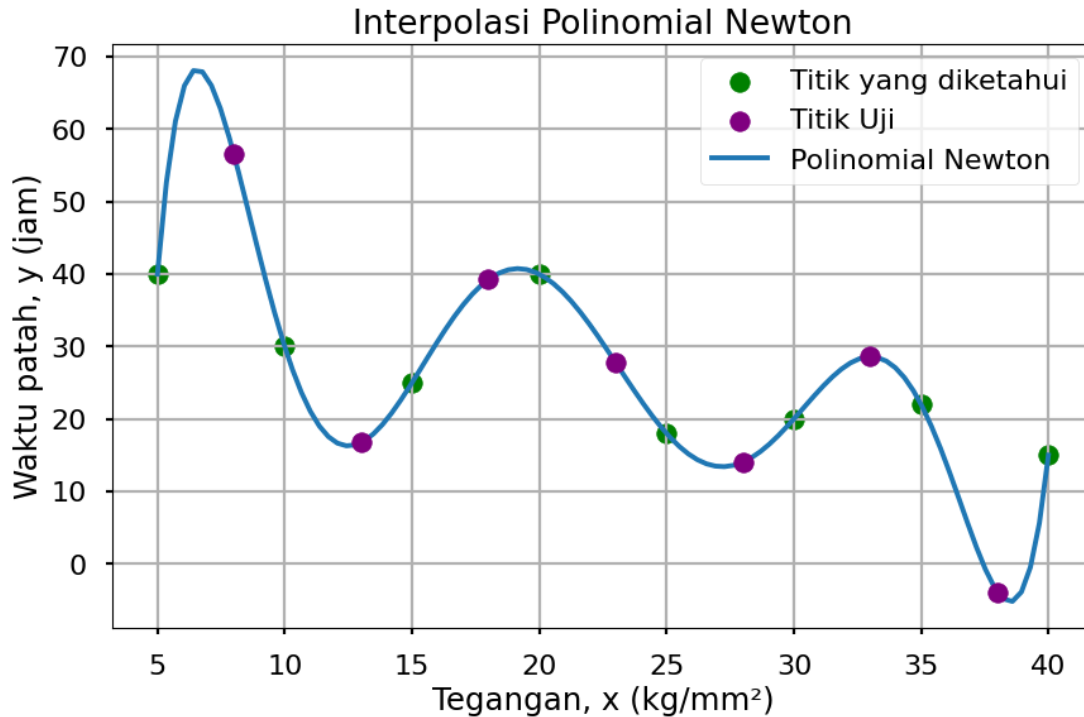
```
plt.figure(figsize=(10, 6))

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
            Newton')
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`.

III. Analisis Hasil



Dalam grafik **Interpolasi Polinomial Newton**, 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.