**BAB I**

**PENDAHULUAN**

* 1. **Tujuan**

Dapat menghitung dan membuat program Interpolasi Newton dan Lagrange menggunakan implementasi microsoft Excel dan Bahasa pemrograman python.

* 1. **Rumusan Masalah**

1. Hitung nilai x = 2.5 dengan metode interpolasi jika diketahui tabel sebagai berikut:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 0.0 | 1.0 | 2.0 | 3.0 | 4.0 |
| f(x) | 1.0000 | 0.5403 | -0.4161 | -0.9900 | -0.6536 |

1. Hitung nilai x = 323.5 dengan metode interpolasi lagrange jika diketahui tabel sebagai berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 321.0 | 322.8 | 324.2 | 325.0 |
| f(x) | 2.50651 | 2.50893 | 2.51081 | 2.51188 |
|  |  |  |  |  |

**BAB II**

**DASAR TEORI**

Interpolasi polinomial merupakan pencarian polinomial derajad n melalui n+1 titik. Metode yang sering dipakai adalah metode interpolasi Newton dan metode interpolasi Lagrange.

1. **Metode Interpolasi Newton**

Diberikan titik data, , jarak titik x dengan selang yang sama, sebagai

Atau dapat ditulis sebagai

Dengan

Algoritma Interpolasi Newton ke depan:

1. Baca data masukan pasangan x dan y.
2. Berikan nilai x yang dicari y nya.
3. Hitung h.
4. Lakukan inisialisasi:

Sum = y0

R = (x-x0)/h

1. Lakukan iterasi proses berikut untuk i=1 sampai n-1
2. Product = 1
3. Untuk j = 0 sampai i - 1 lakukan perhitungan Product = Product \* (u-j)/(j+1)
4. Lakukan penjumlahan Sum = Sum + Product \*
5. Kembalikan nilai sum sebagai hasil perhitungan
6. **Metode Interpolasi Lagrange**

Diberikan titik data, , jarak titik x dengan selang yang tidak sama. Dengan . Polinom Lagrange diberikan oleh

dengan dalam merupakan derajad order yang mengaproksimasi fungsi diberikan titik data, dan .

Algoritma Interpolasi Lagrange:

1. Baca data masukan pasangan x dan y
2. Lakukan inisialisasi:

Sum =

R = (x-/h

1. Lakukan iterasi proses berikut untuk i=1 sampai n-1
2. Product =
3. Untuk j=0 sampai i-1 lakukan perhitungan Product = Product\*(x-/ (-
4. Lakukan penjumlahan Sum = Sum + Product\*
5. Kembalikan nilai Sum sebagai hasil perhitungan

**BAB III**

**PEMBAHASAN**

**3. 1 Penyelesaian dengan Polinomial Newton**

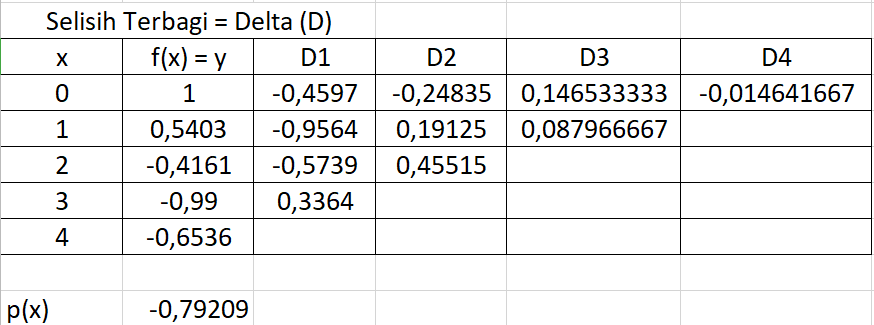
**3. 1. 1 Source Code PolimNewton**

|  |
| --- |
| # Nama: Muhammad Rayhan Magnifica  # NIM: 24060119130065  # Lab: A1  import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  #data pada tabel  x = [0.0, 1.0, 2.0, 3.0, 4.0]  y = [1.0, 0.5403, -0.4161, -0.9900, -0.6536]  # y = f(x)  xinput = 2.5  def newton\_interpolation(x, y, xi):  #Panjang/jumlah data point  n = len(x)  #inisialisasi selisih terbagi  fdd = [[None for x in range(n)] for x in range(n)]  #nilai f(x) pada derajat yang berbeda  yint = [None for x in range(n)]  #nilai error  ea = [None for x in range(n)]    #menemukan perbedaan pembagi  for i in range(n):  fdd[i][0] = y[i]  for j in range(1,n):  for i in range(n-j):  fdd[i][j] = (fdd[i+1][j-1] - fdd[i][j-1])/(x[i+j]-x[i])    fdd\_table = pd.DataFrame(fdd)  print(fdd\_table)    #interpolasi xinput  xterm = 1  yint[0] = fdd[0][0]  for order in range(1, n):  xterm = xterm \* (xi - x[order-1])  yint2 = yint[order-1] + fdd[0][order]\*xterm  ea[order-1] = yint2 - yint[order-1]  yint[order] = yint2    return map(lambda yy, ee : [yy, ee], yint, ea)  a = newton\_interpolation(x, y, xinput)  df = pd.DataFrame(a, columns=['f(x)','error'])  print(  )  print(df) |

**3. 1. 2 Hasil Run Code PolimNewton**

|  |
| --- |
|  |

1. **1. 3 Table Excel PolimNewton**

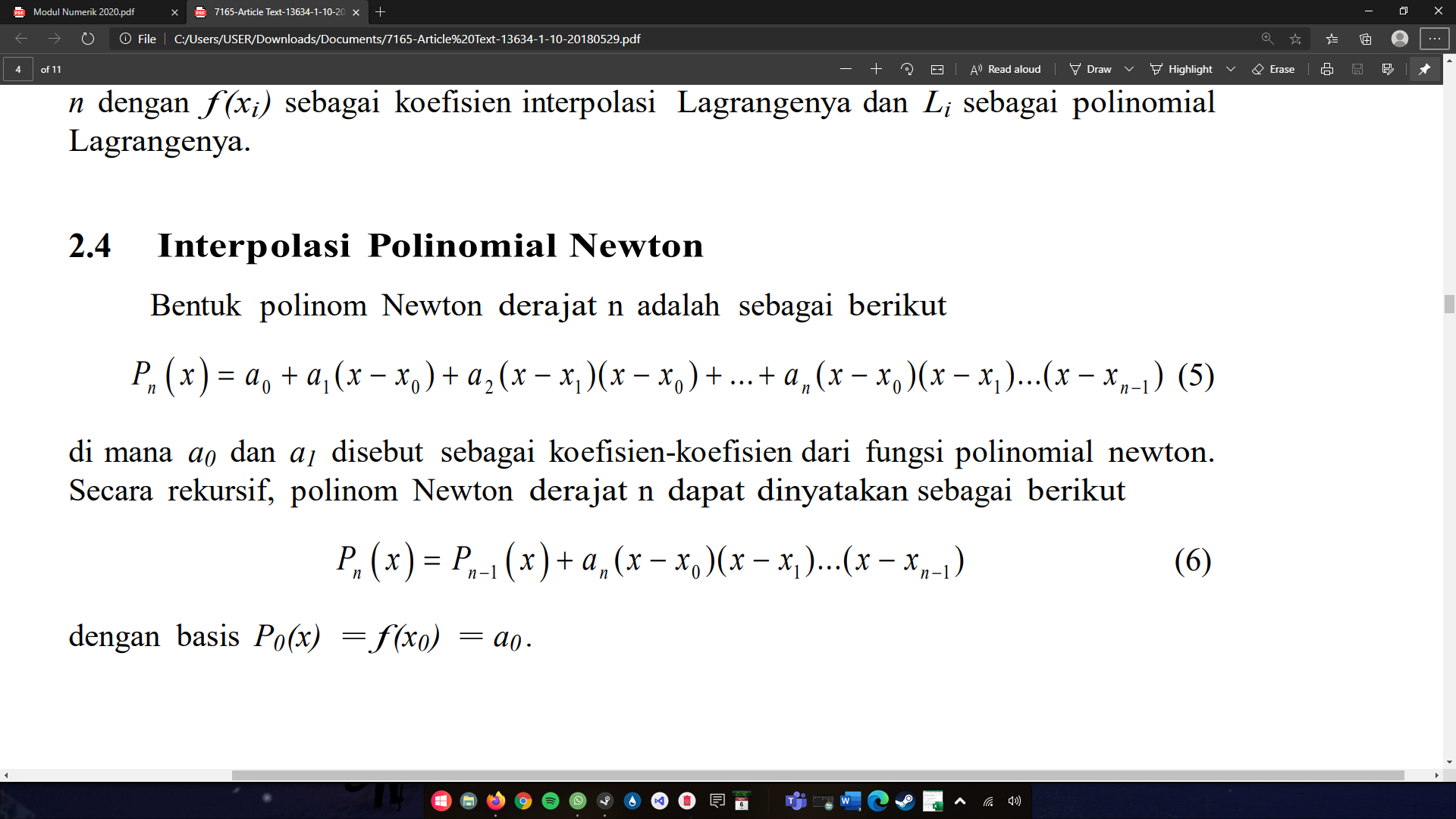


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 0.0 | 1.0 | 2.0 | 2.5 | 3.0 | 4.0 |
| f(x) | 1.0000 | 0.5403 | -0.4161 | -0,79209 | -0.9900 | -0.6536 |

**3. 1. 4 Penjelasan**

Program diatas akan menampilkan solusi dari interpolasi polynomial metode Newton. Karena menggunakan pandas, maka program tersebut sengaja dibuat dengan tampilan akhir berupa tabel. Tabel yang pertama menampilkan Interpolasi Newton lengkap dengan Selisih Terbagi nya. Tabel yang kedua menampilkan hasil P3(x) nya dengan nilai errornya.

Source code diatas merupakan program untuk mencari hasil interpolasi untuk x = 2.5 menggunakan data yang disediakan. Pertama, kita menentukan nilai delta y nya terlebih dahulu dalam bentuk tabel. Kemudian nilai yang dipakai yaitu nilai delta y yang berada pada baris pertama tabel. Lalu dilakukan perhitungan dengan iterasi sehingga ditemukan hasil interpolasinya.



Dimana perulangan akan berada pada range 1 sampai n. Hasil akhir dari program ini adalah **-0,792086**

Implementasi pada Microsoft Excel dirasa lebih mudah karena tidak perlu mengkonversi rumus yang ada menjadi Bahasa pemrograman. Adapun rumus-rumus yang ada sudah saya sertakan diatas. Terdapat sedikit perbedaan pembulatan antara Microsoft Excel dan Bahasa Pemrograman Python. Tapi hal ini tidak mengubah garis besar hasil jawabnya.

**3. 2 Penyelesaian dengan Polinomial Lagrange**

**3. 2. 1 Source Code PolimLagrange**

|  |
| --- |
| # Nama: Muhammad Rayhan Magnifica  # NIM: 24060119130065  # Lab: A1  import numpy as np  # Memasukkan jumlah data yang akan dimasukkan  n = int(input('Masukkan jumlah data: '))  # Making numpy array of n & n x n size and initializing  # to zero for storing x and y value along with differences of y  x = np.zeros((n))  y = np.zeros((n))  # Memasukkan data yang dijadikan acuan untuk polinomial  print('Masukkan data untuk x dan y: ')  for i in range(n):  x[i] = float(input('x['+str(i)+']'))  y[i] = float(input('y['+str(i)+']'))  # Memasukkan nilai interpolasi yang dicari  xp = float(input('Masukkan nilai interpolasi yang dicari: '))  # Inisialisasi nilai hasil interpolasi  Sum = 0  # Metode Interpolasi Lagrange  for i in range(n):  p = 1  for j in range(n):  if i != j:  p = p \* (xp-x[j]) / (x[i]-x[j])  Sum = Sum + p \* y[i]  # Mengeluarkan hasil interpolasi  print('Hasil interpolasi untuk x = %f is %f.' % (xp, Sum)) |

**3. 2. 2 Hasil Run Code PolimLagrange**

|  |
| --- |
|  |

**3. 2.3 Table Excel PolimLagrange**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 321.0 | 322.8 | 323.5 | 324.2 | 325.0 |
| f(x) | 2.50651 | 2.50893 | 2.509871 | 2.51081 | 2.51188 |

**3. 2. 4 Penjelasan**

Program ini dimulai dengan melakukan inisialisasi terhadap sum dimana hasilnya merupakan penjumlahan dari semua product yang akan terjadi. Digunakan variabel n untuk menginisialisasi jumlah perulangan. Body utama program sendiri merupakan implementasi rumus dari interpolasi Metode Lagrange.

Source code diatas merupakan program untuk mencari interpolasi untuk x = 323.5. Pertama, kita memasukkan jumlah data yang akan menjadi acuan kita dalam menentukan persamaan polinomial. Kedua, memasukkan data yang akan dijadikan acuan dalam mencari persamaan yaitu nilai x dan y nya. Ketiga, memasukkan nilai x yang akan dicari hasil polinomialnya. Keempat, program akan melakukan proses perhitungan menggunakan metode

Setelah ditemukan nilai baginya, maka hasil tersebut akan dikalikan dengan product (dalam tabel soal merupakan f(x)) dimana nilai product akan terus berubah sesuai dengan perulangan yang ada. Setelah semua product ditemukan, maka hasilnya akan dijumlahkan sesuai dengan rumus awal

Seperti program sebelumnya, program ini hanya mengkoversi rumus yang ada agar dapat dipahami komputer dengan Bahasa pemrograman python.

**BAB IV**

**PENUTUP**

**4.1 Kesimpulan**

Nilai dari suatu fungsi *x* dapat dicari dengan menggunakan Interpolasi Newton dan Interpolasi Lagrange. Langkah-langkah yang tersedia dapat dibuat dengan menggunakan Pemrograman Bahasa *Python* dan Microsoft Excel sehingga prosesnya menjadi lebih sederhana dan tidak perlu melakukan iterasi secara manual menggunakan baik menggunakan kalkulator maupun perhitungan-perhitungan manual lain yang kiranya akan menghabiskan banyak waktu untuk perhitungannya.

1. Pada soal pertama tentang implementasi Metode *Interpolation Newton Method,* dicari hasil persamaan f(x) dengan nilai x adalah 2.5. Diketahui tabel

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 0.0 | 1.0 | 2.0 | 2.5 | 3.0 | 4.0 |
| f(x) | 1.0000 | 0.5403 | -0.4161 | -0,79209 | -0.9900 | -0.6536 |

Hasil akhirnya adalah **-0.79209** (menggunakan Microsoft Excel) dan **-0.792086** menggunakan program Bahasa Python. Ada sedikit perbedaan pembulatan. Namun hal ini tidak mengubah garis besar hasil jawabnya.

1. Pada soal pertama tentang implementasi Metode *Interpolation Newton Method,* dicari hasil persamaan f(x) dengan nilai x adalah 323.5. Diketahui tabel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 321.0 | 322.8 | 323.5 | 324.2 | 325.0 |
| f(x) | 2.50651 | 2.50893 | 2.509871 | 2.51081 | 2.51188 |

Dengan menggunakan Bahasa pemrograman Python, diketahui nilai akhirnya adalah **2.509870836884469.**

**DAFTAR PUSTAKA**

* Universitas Diponegoro. 2020. *Modul Numerik 2020*: Universitas Diponegoro
* Medium, 29 Mei 2020, Newton’s Divided-difference and Lagrange interpolating polynomials Python implementation, Diakses pada 7 November 2020, dari https://medium.com/analytics-vidhya/newtons-divided-difference-and-lagrange-interpolating-polynomials-python-implementation-6dbf35376b2b.