Praktikum Fisika Komputasi Senin, 7 Oktober 2024

Integral Metode Numerik

Berdasarkan soal yang diberikan yaitu menyelesaikan soal integral dengan cara yang berbeda menggunakan pemrograman Python, maka telah didapatkan hasilnya yang tertera pada pemrograman Python saat di-run.

Untuk hasil yang didapat dari masing-masing metode, untuk Metode Numerik sendiri hasilnya didapat dari hasil perhitungan manual yang mendapatkannya itu dengan cara menghitung integralnya secara terpisah yang pertama dihitung dulu integral x^-3 nya lalu kemudian dihitung integral cos (x) dx nya lalu jika hasilnya sudah didapat baru dijumlahkan kedua integral itu . Untuk metode Trapezoid sendiri hasilnya didapat dengan cara fungsi x^-3 dievaluasi pada sejumlah titik di interval (1, 5) yang dibagi menjadi 10.000 bagian yang hasilnya didapatkan akan mendekati nilai eksak dan kemudian didapatkan visualisasi data berupa grafik fungsi yang diintegrasikan dan pembagian trapezoid yang digunakan untuk menghitung integral. Untuk metode Simpson sendiri didapat dengan cara setiap kelompok 3 titik dihubungkan oleh parabola yang digunakan untuk menghitung area kurva lalu kemudian hasil yang didapat lebih mendekati metode eksak dibandingkan dengan metode Trapezoid kemudian didapatkan visualisasi data berupa grafik kurva fungsi yang diintegralkan tetapi untuk metode Simpson ini, hasil yang didapatkan lebih halus dalam mendekati area bawah kurva.

Berdasarkan hasil integral yang telah didapat maka dari segi perbedaan, untuk metode Eksak sendiri terbilang akurat, tetapi untuk sebagian kasus tidak dapat diselesaikan dengan mudah, untuk metode Trapezoid sendiri terbilang sederhana dan mudah digunakan tetapi kurang akurat untuk fungsi non-linear, untuk metode Simpson sendiri bisa dibilang lebih akurat dibandingkan metode Trapezoid karena menggunakan parabola untuk mendekati area dibawah kurva sehingga lebih efektif dan akurat untuk menghitung fungsi non-linear. Jika melihat dari segi keefektifan maka dapat disimpulkan untuk metode Simpson sendiri lebih akurat untuk digunakan karena mendekati kurva yang tidak linear.

Kode Program

```
# Mengimport Library
import numpy as np
import matplotlib.pyplot as plt
```

```
# Integral
def func(x):
    return (x**-3) + np.cos(x)
a = 1.0
b = 5.0
```

```
# Metode Trapezoid
n = 10000
dx = (b-a)/(n-1)
x = np.linspace(a,b,n)

sigma = 0
for i in range (1, n-1):
    sigma += func(x[i])

hasil = 0.5*dx*(func(x[0])+2*sigma+func(x[-1]))

print(hasil)
```

```
xp =np.linspace(a,b,10000)
plt.plot(xp,func(xp))
plt.show()
```

```
xp =np.linspace(a,b,10000)
plt.plot(xp,func(xp))

for i in range (n):
    plt.bar(x[i],func(x[i]),align = 'edge',width = 0.000001, edgecolor='black')

plt.show()
```

```
xp =np.linspace(a,b,10000)
plt.plot(xp,func(xp))

for i in range (n):
    plt.bar(x[i],func(x[i]),align = 'edge',width = 0.000001, edgecolor='black')

plt.fill_between(x,func(x),color= 'blue', alpha=0.5)

plt.show()
```

```
# Mengimport Library
import numpy as np
import matplotlib.pyplot as plt

# Fungsi yang akan diintegralkan
def func(x):
    return (x**-3) + np.cos(x)

# Batas Integrasi
a = 1.0   # Batas Bawah
b = 5.0  # Batas Atas
n = 10000  # Jumlah grid, harus ganjil untuk metode Simpson
```

```
# Simpson's Rule
if n % 2 == 0:
    n += 1

x = np.linspace(a, b, n)
dx = (x[-1] - x[0]) / (n - 1)

hasil = func(x[0]) + func(x[-1])

for i in range(1, n-1, 2):
    hasil += 4 * func(x[i])

for i in range(2, n-2, 2):
    hasil += 2 * func(x[i])

hasil *= dx / 3

#Visualisasi grafik dan bar
xp = np.linspace(a,b,10000)
plt.plot(xp, func(xp))

for i in range(n):
    plt.bar(x[i], func(x[i]), align='edge', width=dx, color='red', edgecolor='black')
plt.show()
print(hasil)
```