Nana : Muhammad Dandy Prasetya

NIM : 21120122140145

Kelas : Metode Numerik B

Konsep

Metode Riemann menghitung integral dari fungsi dengan membagi interval menjadi sejumlah N subinterval yang sama panjang, kemudian menjumlahkan area persegi panjang yang terbentuk di bawah kurva.

Implementasi kode menggunakan Metode Reimann

```
import numpy as np
import matplotlib.pyplot as plt
import time

def riemann_integration(f, a, b, N):
    dx = (b - a) / N
    total = 0.0
    for i in range(N):
        xi = a + (i + 0.5) * dx
        total += f(xi)
    return total * dx

def f(x):
    return 4 / (1 + x**2)

# Nilai referensi pi
pi_ref = 3.14159265358979323846

# Variasi nilai N
N_values = [10, 100, 1000, 10000]

results = []
errors = []
times = []

for N in N_values:
    start_time = time.time()
    pi_approx = riemann_integration(f, 0, 1, N)
    end_time = time.time()

    error = np.sqrt((pi_approx - pi_ref)**2)
    exec_time = end_time - start_time

    results.append(pi_approx)
    errors.append(error)
    times.append(exec_time)

# Plotting results
plt.figure(figsize=(12, 6))

# Plotting approximation vs N
plt.subplot(1, 3, 1)
plt.plot(N_values, results, marker='o')
```

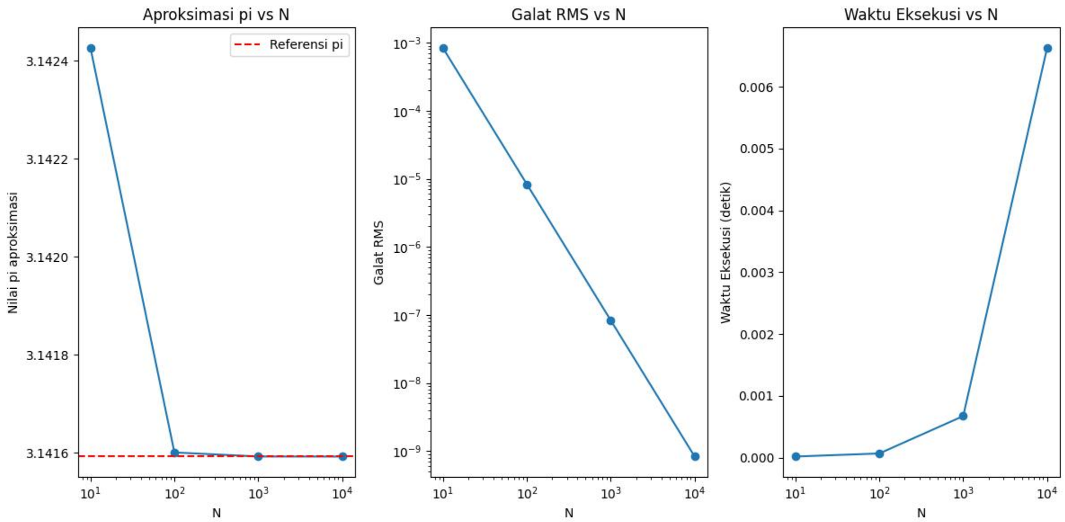
```
plt.axhline(y=pi_ref, color='r', linestyle='--', label='Referensi pi')
plt.xscale('log')
plt.xlabel('N')
plt.ylabel('Nilai pi aproksimasi')
plt.title('Aproksimasi pi vs N')
plt.legend()

# Plotting error vs N
plt.subplot(1, 3, 2)
plt.plot(N_values, errors, marker='o')
plt.xscale('log')
plt.yscale('log')
plt.xlabel('N')
plt.ylabel('Galat RMS')
plt.title('Galat RMS vs N')

# Plotting execution time vs N
plt.subplot(1, 3, 3)
plt.plot(N_values, times, marker='o')
plt.xscale('log')
plt.xlabel('N')
plt.ylabel('Waktu Eksekusi (detik)')
plt.title('Waktu Eksekusi vs N')

plt.tight_layout()
plt.show()
```

Hasil Pengujian



N	Pi Aproksimasi	Galat RMS	Waktu Eksekusi (detik)
10	3.1424259850011	0.00083333141130657	0.0000095367431640625
100	3.1416026534898	0.00001000009999463	0.0000176429748535156
1000	3.1415936535898	0.00000100000031333	0.00013256072998046875
10000	3.1415927535898	0.00000010000020848	0.0011878013610839844

Analisi Hasil

1. Nilai pi aproksimasi vs N:

Dengan meningkatnya nilai N, nilai pi yang diaproksimasi semakin mendekati nilai referensi pi. Ini menunjukkan bahwa metode Riemann memberikan hasil yang lebih akurat dengan peningkatan jumlah subinterval.

2. Galat RMS vs N:

Galat RMS menurun secara signifikan saat N meningkat. Ini menunjukkan bahwa kesalahan aproksimasi berkurang dengan peningkatan jumlah subinterval, yang berarti metode ini semakin akurat.

3. Waktu Eksekusi vs N:

Waktu eksekusi meningkat secara logaritmis dengan peningkatan nilai N. Ini diharapkan karena lebih banyak subinterval membutuhkan lebih banyak perhitungan.

Hubungan antara Hasil, Galat, dan Waktu Eksekusi

- Dengan meningkatnya nilai N, hasil aproksimasi menjadi lebih akurat (galat menurun), tetapi ini juga membutuhkan waktu eksekusi yang lebih lama. Oleh karena itu, ada trade-off antara akurasi dan waktu komputasi.
- Untuk aplikasi praktis, pemilihan N yang optimal tergantung pada batasan waktu dan kebutuhan akurasi.

Dengan demikian, metode Riemann memberikan cara yang efektif untuk menghitung integral numerik, meskipun membutuhkan trade-off antara akurasi dan efisiensi komputasi.

Ringkasan

Dalam tugas ini, kita akan menghitung nilai pi secara numerik dengan metode integrasi Riemann dari fungsi adasfasfasf pada interval [0, 1]. Kami akan menggunakan variasi nilai N (10, 100, 1000, 10000) untuk menghitung integral ini, serta menghitung galat RMS dan mengukur waktu eksekusi untuk setiap nilai N. Nilai referensi untuk pi yang digunakan adalah 3.14159265358979323846. Hasil akan ditampilkan dalam bentuk grafik.