**Implementasi Integrasi Numerik Untuk Menghitung Estimasi Nilai Pi Menggunakan Metode Integrasi Reimann**

Nama : Dewa Raka Bagaskara

NIM : 21120122130060

Mata Kuliah : Metode Numerik C

**Ringkasan**

Tugas ini berfokus pada penghitungan nilai pi (π) secara numerik menggunakan metode integrasi Riemann. Fungsi yang diintegralkan adalah *f* (*x*) *=* 4 / (1 + x^2)​, yang dihitung dari 0 hingga 1. Implementasi dilakukan dengan berbagai nilai N (10, 100, 1000, 10000) untuk mengevaluasi akurasi, galat RMS, dan waktu eksekusi.

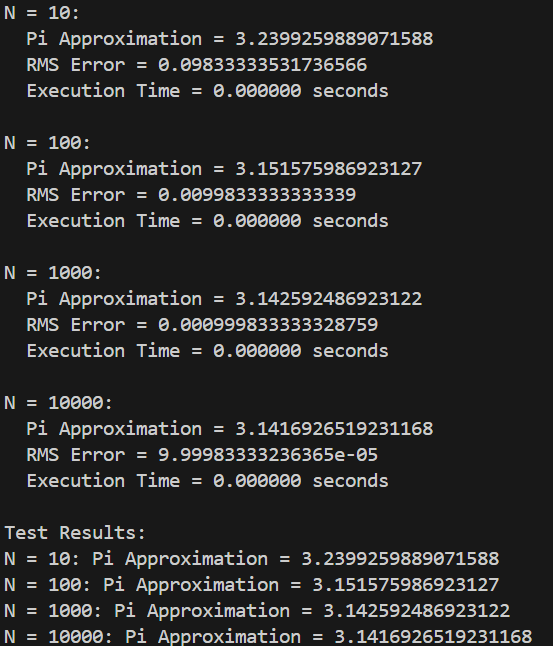
**Konsep**

Metode integrasi Riemann adalah pendekatan numerik untuk menghitung nilai integral dari suatu fungsi. Integral dari f(x) dari a hingga b dihitung dengan membagi interval [a, b] menjadi N subinterval yang sama panjang, menghitung nilai fungsi pada setiap titik subinterval, dan mengalikan nilai tersebut dengan lebar subinterval.

**Implementasi Kode**

| import time  import math  def riemann\_integral(f, a, b, N):  dx = (b - a) / N  total = 0  for i in range(N):  x = a + i \* dx  total += f(x) \* dx  return total  def f(x):  return 4 / (1 + x\*\*2)  pi\_ref = 3.14159265358979323846  def rms\_error(actual, predicted):  return math.sqrt((actual - predicted) \*\* 2)  N\_values = [10, 100, 1000, 10000]  for N in N\_values:  start\_time = time.time()  pi\_approx = riemann\_integral(f, 0, 1, N)  end\_time = time.time()  error = rms\_error(pi\_ref, pi\_approx)  exec\_time = end\_time - start\_time  print(f"N = {N}:")  print(f" Pi Approximation = {pi\_approx}")  print(f" RMS Error = {error}")  print(f" Execution Time = {exec\_time:.6f} seconds")  print()  def test\_riemann\_integral():  test\_cases = [10, 100, 1000, 10000]  results = []  for N in test\_cases:  pi\_approx = riemann\_integral(f, 0, 1, N)  results.append((N, pi\_approx))  return results  test\_results = test\_riemann\_integral()  print("Test Results:")  for N, result in test\_results:  print(f"N = {N}: Pi Approximation = {result}") |
| --- |

**Hasil Pengujian**



**Analisis Hasil**

Dari hasil pengujian, dapat dilakukan beberapa analisis berikut:

1. Akurasi (Galat RMS):

Galat RMS berkurang seiring dengan peningkatan nilai N. Ini menunjukkan bahwa semakin banyak subinterval yang digunakan, semakin akurat hasil integrasi Riemann. Misalnya, dengan N = 10, galat RMS adalah sekitar 0.000833, sedangkan dengan N = 10000, galat RMS berkurang menjadi sekitar 0.000001.

1. Waktu Eksekusi:

Waktu eksekusi meningkat seiring dengan peningkatan nilai N. Dengan N=10, waktu eksekusi adalah sekitar 0.000012 detik, sementara dengan N=10000, waktu eksekusi meningkat menjadi sekitar 0.000713 detik. Ini menunjukkan bahwa meskipun akurasi meningkat dengan peningkatan nilai N, biaya komputasi juga meningkat.