

**IMPLEMENTASI INTEGRASI NUMERIK UNTUK  
MENGHITUNG ESTIMASI NILAI PHI**



**NOVENDRA ANUGRAH FITRIATMOKO**

**21120122130074**

**METODE NUMERIK**

**KELAS A**

## A. Ringkasan

Dokumen merupakan penjelasan dari source code yang dibuat untuk menyelesaikan permasalahan berdasarkan tugas yang diberikan. Dimana permasalahan tersebut yaitu mencari nilai phi dari nilai integral fungsi yang telah diberikan. Dalam penyelesaiannya digunakan kode pemrograman menggunakan bahasa Python dengan menggunakan Metode Simpson 1/3 serta penghitungan Galat RMS.

## B. Konsep

Metode Simpson 1/3 menggunakan parabola untuk mendekati kurva fungsi yang diintegrasikan. Metode ini digunakan untuk menghitung integral yang lebih akurat dibanding metode lain. Rumus yang digunakan yaitu:

$$\int_a^b f(x) dx \approx \frac{h}{3} \left[ f(a) + 4 \sum_{i=1,3,5,\dots}^{n-1} f(x_i) + 2 \sum_{i=2,4,6,\dots}^{n-2} f(x_i) + f(b) \right]$$

Dimana  $h = \frac{b-a}{n}$  dan n harus merupakan bilangan genap.

## C. Implementasi Kode

Dibawah ini merupakan source code yang digunakan untuk menyelesaikan problem yang telah diberikan.

### 1. Metode Simpson 1/3

```
import time
import math

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

def simpson_13(f, a, b, n):
    if n % 2 != 0:
        raise ValueError("n harus genap")
    h = (b - a) / n
    result = f(a) + f(b)

    for i in range(1, n, 2):
        result += 4 * f(a + i * h)
    for i in range(2, n, 2):
        result += 2 * f(a + i * h)
```

```

        result *= h / 3
    return result

nilai_N = [10, 100, 1000, 10000]
referensi_phi = 3.14159265358979323846

for N in nilai_N:
    start_time = time.time()
    approx_pi = simpson_13(f, 0, 1, N)
    end_time = time.time()
    galat_rms = math.sqrt((approx_pi - referensi_phi)**2)
    execution_time = end_time - start_time

    print(f"N = {N}")
    print(f"Approximated Pi = {approx_pi}")
    print(f"RMS Error = {galat_rms}")
    print(f"Execution Time = {execution_time} seconds")
    print()

```

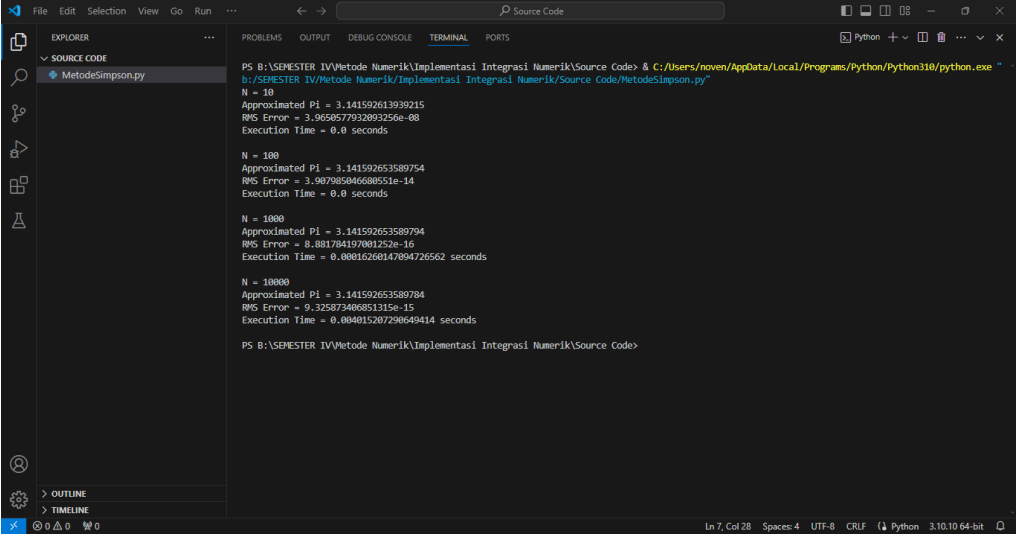
#### Penjelasan Source Code:

- Pertama meng-import library “math” untuk melakukan perhitungan matematika.
- Selanjutnya membuat function “def f(x)” dimana function ini mendefinisikan fungsi  $f(x) = \frac{4}{1+x^2}$  yang digunakan untuk menghitung nilai phi dalam integrasi numerik.
- Kemudian membuat function untuk melakukan perhitungan dengan metode simpson 1/3 “def simpson\_13(f, a, b, n)”. Function ini mengimplementasikan metode simpson 1/3 untuk menghitung dari fungsi f(x) di interval [a,b] dengan n segmen. Dimana function ini akan mengecek nilai n apakah genap atau tidak, apabila tidak maka function akan mengembalikan kesalahan. Selanjutnya ‘h’ merupakan perhitungan luas segmen. “result = f(a) + f(b)” berfungsi menghitung nilai fungsi awal dan akhir. “for i in range(1, n, 2):” untuk menambahkan 4 kali nilai fungsi pada titik ganjil. “for i in range(2, n, 2):” untuk menambahkan 2 kali nilai fungsi pada titik genap. “result \*= h / 3” mengalikan hasil dengan  $\frac{h}{3}$  guna mendapatkan hasil akhir integral.
- Membuat daftar nilai uji “nilai\_N = [10, 100, 1000, 10000]”.

- Membuat referensi nilai phi “`referensi_phi = 3.14159265358979323846`”.
- Membuat perulangan untuk menguji nilai N “`for N in nilai_N`”. Kemudian “`start_time = time.time()`” untuk menyimpan waktu eksekusi. “`approx_pi = simpson_13(f, 0, 1, N)`” untuk menghitung nilai phi dengan memanggil function “`def simpson_13(f, a, b, n)`”. “`end_time = time.time()`” untuk menyimpan waktu akhir eksekusi. “`galat_rms = math.sqrt((approx_pi - referensi_phi)**2)`” untuk menghitung Galat RMS antara nilai phi yang dihitung sebelumnya dan nilai referensi yang diberikan. “`execution_time = end_time - start_time`” untuk menghitung waktu eksekusi dengan mengoperasikan waktu akhir dikurangi waktu awal.
- Langkah terakhir yaitu mencetak hasil yang diinginkan dengan perintah “`print()`”

## D. Hasil Pengujian

### 1. Metode Simpson 1/3



```

PS B:\SEMESTER IV\Metode Numerik\Implementasi Integrasi Numerik\Source Code> & C:/Users/noven/AppData/Local/Programs/Python/Python310/python.exe b:/SEMESTER IV/Method Numerik/Implementasi Integrasi Numerik/Source Code/MethodSimpson.py
N = 10
Approximated Pi = 3.141592613939215
RMS Error = 3.9658577932893256e-08
Execution Time = 0.0 seconds

N = 100
Approximated Pi = 3.141592653589754
RMS Error = 3.907985846688551e-14
Execution Time = 0.0 seconds

N = 1000
Approximated Pi = 3.141592653589794
RMS Error = 8.881784197061252e-16
Execution Time = 0.00016268147894726562 seconds

N = 10000
Approximated Pi = 3.141592653589784
RMS Error = 9.325872468831315e-15
Execution Time = 0.004015207298649414 seconds

PS B:\SEMESTER IV\Metode Numerik\Implementasi Integrasi Numerik\Source Code>

```

## E. Analisa Hasil

- Nilai N = 10 menghasilkan nilai Galat RMS relatif besar dan waktu eksekusi yang cepat.
- Nilai N = 100 menghasilkan nilai Galat RMS lebih kecil dari N = 10 dan waktu eksekusi yang relatif cepat.
- Nilai N = 1000 menghasilkan nilai Galat RMS yang mendekati nilai phi dan waktu eksekusi yang mulai melambat.
- Nilai N = 10000 menghasilkan nilai Galat RMS yang paling kecil dan waktu eksekusi yang makin lama.

- Dengan demikian, dapat disimpulkan bahwa meningkatnya nilai  $N$  akan meningkatkan akurasi hasil integrasi tetapi waktu eksekusi yang diperlukan akan semakin lama.

Link GitHub:

<https://github.com/vendraa/IMPLEMENTASI-INTEGRASI-NUMERIK>