

Implementasi Metode Trapezoida untuk Menghitung Jarak Tempuh Benda Berdasarkan Data Kecepatan terhadap Waktu Menggunakan Python

Radit Ryan Nugroho¹, Uci Ulan Dari², Ajizah Hanum Lubis³, Muhammad Ammar Yasir Adrisa⁴, Putra Agrifa Pasaribu⁵, Neyna Alisya Sari Br Barus⁶, Habibi Azka Nasution⁷

^{1,2,3,4,5,6,7} Department of Physics, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Medan, Medan, North Sumatra, Indonesia
e-mail: habibiazka@unimed.ac.id

Abstrak

Metode integrasi numerik merupakan teknik yang digunakan untuk menyelesaikan persoalan integral yang sulit dihitung secara analitik. Penelitian ini bertujuan menerapkan metode trapezoida pada kasus fisika berupa perhitungan jarak tempuh kendaraan berdasarkan hubungan kecepatan terhadap waktu menggunakan bahasa pemrograman Python. Pengujian dilakukan pada dua jenis data, yaitu fungsi kecepatan kontinu dan data kecepatan diskrit hasil pengamatan. Implementasi memanfaatkan library NumPy, Matplotlib, dan SciPy untuk mendukung komputasi, visualisasi, serta validasi hasil perhitungan. Tingkat akurasi metode dianalisis melalui error absolut, error relatif, dan konvergensi numerik dengan variasi jumlah interval. Hasil penelitian menunjukkan bahwa peningkatan jumlah interval menyebabkan error semakin kecil sehingga hasil integrasi numerik mendekati solusi analitik. Pada fungsi kontinu diperoleh hasil integral yang mendekati nilai eksak sebesar 155,00, sedangkan pada data diskrit diperoleh estimasi jarak tempuh sebesar 95 meter. Validasi menggunakan fungsi quad() pada SciPy menghasilkan nilai integral yang identik dengan solusi analitik. Perbandingan dengan metode Simpson menunjukkan akurasi yang lebih tinggi, namun metode trapezoida lebih unggul dalam kesederhanaan algoritma dan kemudahan implementasi. Penelitian ini menunjukkan bahwa metode trapezoida efektif diterapkan pada fungsi kontinu maupun data diskrit dalam perhitungan jarak tempuh kendaraan berbasis Python. Metode ini juga memberikan hasil stabil, mudah dipahami, serta relevan digunakan dalam pembelajaran komputasi numerik dan analisis fisika terapan secara sederhana.

Kata Kunci: Integrasi Numerik, Metode Trapezoida, Fungsi Kontinu, Data Diskrit, Jarak Tempuh Kendaraan, Python.

Abstract

Numerical integration methods are techniques used to solve integral problems that are difficult to calculate analytically. This study aims to apply the trapezoidal method to a physics problem involving the calculation of a vehicle's distance traveled based on the relationship between speed and time using the Python programming language. Testing was conducted on two types of data: continuous speed functions and discrete speed data from observations. The implementation utilizes the NumPy, Matplotlib, and SciPy libraries to support computation, visualization, and validation of the calculation results. The method's accuracy was analyzed through absolute error, relative error, and numerical convergence with varying numbers of intervals. The results show that increasing the number of intervals reduces the error, causing the numerical integration results to approach the analytical solution. For the continuous function, the integral result is close to the exact value of 155.00, while for the discrete data, the estimated distance traveled is 95 meters. Validation using the quad() function in SciPy yields an integral value identical to the analytical solution. A comparison with Simpson's method shows higher accuracy, but the trapezoidal method is superior in terms of algorithmic simplicity and ease of implementation. This study demonstrates that the trapezoidal method is effective for both continuous functions and discrete data in Python-based vehicle mileage calculations. This method also yields stable results, is easy to understand, and is suitable for use in teaching numerical computation and applied physics analysis in a straightforward manner.

Keywords: Numerical Integration, Trapezoidal Rule, Continuous Function, Discrete Data, Vehicle Travel Distance, Python.

LATAR BELAKANG

Perkembangan teknologi komputasi telah memberikan kontribusi besar dalam penyelesaian berbagai persoalan matematika, fisika, teknik, dan analisis data. Dalam banyak kasus, solusi analitik terhadap suatu permasalahan matematika sulit diperoleh karena bentuk fungsi yang kompleks atau keterbatasan metode analitik yang tersedia. Oleh karena itu, metode numerik digunakan sebagai pendekatan alternatif untuk memperoleh solusi hampiran dengan tingkat akurasi tertentu [1], [2].

Salah satu cabang penting dalam metode numerik adalah integrasi numerik. Integrasi numerik digunakan untuk menghitung luas daerah di bawah kurva ketika penyelesaian integral secara eksak sulit dilakukan atau data yang tersedia berbentuk diskrit [3]. Metode ini banyak diterapkan dalam bidang fisika, teknik elektro, pengolahan sinyal, mekanika fluida, statistika, hingga analisis data eksperimen [4].

Metode trapezoida merupakan salah satu metode integrasi numerik yang paling sederhana dan banyak digunakan karena memiliki algoritma yang mudah diterapkan serta efisiensi komputasi yang baik [5]. Metode ini bekerja dengan membagi interval integral menjadi beberapa bagian kecil kemudian mendekati luas daerah di bawah kurva

menggunakan sejumlah trapesium. Semakin kecil panjang interval yang digunakan maka hasil pendekatan integral akan semakin mendekati solusi analitik [6].

Penelitian mengenai metode integrasi numerik telah dilakukan oleh berbagai peneliti. Chapra dan Canale [1] menjelaskan bahwa metode trapezoida memiliki error orde kuadratik terhadap panjang interval sehingga akurasi meningkat ketika jumlah interval diperbesar. Burden dan Faires [2] menunjukkan bahwa metode trapezoida efektif digunakan pada fungsi kontinu dengan perubahan gradien yang relatif stabil. Penelitian Kumar dan Singh [7] juga menunjukkan bahwa implementasi Python dapat meningkatkan efisiensi komputasi numerik serta mempermudah visualisasi hasil integrasi.

Meskipun demikian, meskipun penelitian mengenai metode trapezoida telah banyak dilakukan, sebagian besar penelitian masih menggunakan fungsi matematika ideal berbentuk polinomial sederhana dan belum banyak mengaitkannya dengan kasus fisika nyata berbasis data. Oleh karena itu, penelitian ini menerapkan metode trapezoida pada kasus perhitungan jarak tempuh benda berdasarkan hubungan kecepatan terhadap waktu. Selain menghitung integral numerik, penelitian ini juga menganalisis konvergensi error, membandingkan hasil dengan metode Simpson dan library SciPy, serta memvisualisasikan proses integrasi menggunakan Python. Selain itu, visualisasi hubungan antara jumlah interval dan tingkat akurasi masih jarang dibahas secara sistematis. Oleh karena itu, penelitian ini dilakukan untuk mengevaluasi hubungan kuantitatif antara jumlah interval dan penurunan error pada metode trapezoida menggunakan pendekatan komputasi berbasis Python.

Kebaruan penelitian ini terletak pada integrasi antara implementasi metode trapezoida berbasis Python, analisis konvergensi numerik terhadap variasi interval, evaluasi error otomatis, serta visualisasi grafik integrasi numerik secara sistematis. Penelitian ini tidak hanya menampilkan hasil integral numerik, tetapi juga mengevaluasi hubungan matematis antara jumlah interval dan tingkat akurasi metode trapezoida.

Python dipilih dalam penelitian ini karena menyediakan berbagai pustaka numerik yang efisien seperti NumPy dan Matplotlib yang mendukung komputasi ilmiah serta visualisasi data secara optimal [8]. Selain bersifat open-source, Python juga memiliki sintaks yang sederhana sehingga mudah digunakan dalam pengembangan algoritma numerik. Penelitian ini bertujuan untuk mengimplementasikan metode trapezoida menggunakan Python dalam menghitung luas daerah di bawah kurva serta menganalisis pengaruh jumlah interval terhadap error integrasi numerik.

METODE

2.1 Metode trapezoida

Penelitian ini menggunakan metode integrasi numerik trapezoida untuk menghitung jarak tempuh kendaraan berdasarkan hubungan kecepatan terhadap waktu dengan bantuan bahasa pemrograman Python. Metode trapezoida dipilih karena memiliki konsep sederhana, mudah diimplementasikan, dan mampu memberikan pendekatan integral numerik yang cukup baik.

Penelitian dilakukan melalui beberapa tahapan, yaitu penentuan fungsi kecepatan, penentuan batas interval, pembagian interval, implementasi algoritma metode trapezoida

menggunakan Python, visualisasi grafik, serta analisis error hasil perhitungan. Tahapan tersebut digunakan agar proses perhitungan numerik dapat dilakukan secara sistematis dan hasilnya dapat dibandingkan dengan solusi analitik.

Kasus fisika yang digunakan dalam penelitian ini adalah perhitungan jarak tempuh kendaraan berdasarkan fungsi kecepatan terhadap waktu. Selain menggunakan fungsi kecepatan kontinu, penelitian ini juga mempertimbangkan data kecepatan diskrit yang merepresentasikan hasil pengamatan kendaraan pada interval waktu tertentu. Data diskrit digunakan untuk mensimulasikan kondisi nyata, di mana data kecepatan umumnya diperoleh dari sensor kendaraan atau hasil pengamatan lapangan. Penggunaan kedua jenis data tersebut bertujuan untuk mengevaluasi kemampuan metode trapezoida dalam menghitung jarak tempuh baik pada fungsi matematis maupun pada data hasil pengukuran.

$$v(t) = 3t^2 + 2t + 1$$

dengan interval waktu:

$$0 \leq t \leq 5$$

Dalam fisika, integral fungsi kecepatan terhadap waktu digunakan untuk menghitung jarak tempuh benda. Oleh karena itu, jarak tempuh kendaraan dapat dinyatakan sebagai:

$$s = \int_0^5 v(t) dt$$

Sehingga diperoleh:

$$s = \int_0^5 (3t^2 + 2t + 1) dt$$

Hasil integral analitik dari fungsi tersebut adalah:

$$s_{\text{analitik}} = 155$$

Metode trapezoida membagi interval integral menjadi beberapa bagian dengan panjang interval yang sama. Panjang interval dihitung menggunakan persamaan:

$$h = \frac{b - a}{n}$$

dengan:

- h = panjang interval,
- a = batas bawah,
- b = batas atas,
- n = jumlah interval.

Titik pembagian interval dinyatakan sebagai:

$$x_i = a + ih$$

Metode trapezoida digunakan untuk menghitung pendekatan integral numerik menggunakan persamaan:

$$I \approx \frac{h}{2} \left[f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n) \right]$$

Persamaan tersebut menunjukkan bahwa titik-titik di bagian tengah interval memiliki kontribusi dua kali lebih besar dibandingkan titik awal dan titik akhir.

2.2 Analisis Error

Analisis error digunakan untuk mengetahui tingkat ketelitian metode trapezoida terhadap solusi analitik. Dengan adanya analisis error, hasil perhitungan numerik tidak hanya dinilai berdasarkan besar nilai integralnya, tetapi juga berdasarkan seberapa dekat hasil tersebut terhadap nilai eksak.

Error absolut dihitung menggunakan persamaan:

$$E_a = |I_{\text{analitik}} - I_{\text{numerik}}|$$

Semakin kecil nilai error absolut, maka hasil pendekatan numerik semakin akurat. Error relatif digunakan untuk mengetahui tingkat kesalahan dalam bentuk persentase:

$$E_r = \left| \frac{I_{\text{analitik}} - I_{\text{numerik}}}{I_{\text{analitik}}} \right| \times 100\%$$

Konvergensi metode trapezoida terhadap solusi analitik dinyatakan sebagai:

$$\lim_{n \rightarrow \infty} I_{\text{numerik}} = I_{\text{analitik}}$$

Metode trapezoida memiliki orde error global:

$$O(h^2)$$

Hal ini menunjukkan bahwa semakin kecil panjang interval, maka error metode trapezoida akan semakin kecil sehingga hasil numerik mendekati solusi eksak.

2.3 Metode Pembandingan

Sebagai pembandingan, penelitian ini juga menggunakan metode Simpson dan validasi menggunakan library SciPy. Metode Simpson digunakan karena memiliki tingkat akurasi yang lebih tinggi dibandingkan metode trapezoida pada banyak kasus integrasi numerik, terutama ketika fungsi yang digunakan relatif halus.

Persamaan metode Simpson dinyatakan sebagai:

$$\int_a^b f(x) dx \approx \frac{h}{3} [f(x_0) + 4 \sum f(x_{\text{ganjil}}) + 2 \sum f(x_{\text{genap}}) + f(x_n)]$$

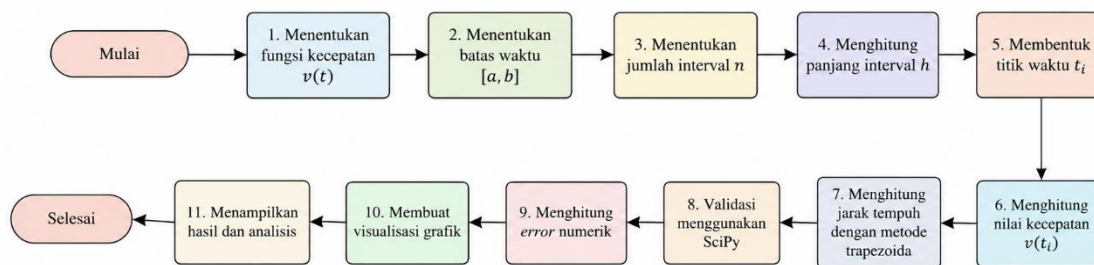
Selain itu, validasi hasil integral dilakukan menggunakan library SciPy melalui fungsi `quad()` untuk memperoleh solusi numerik dengan tingkat akurasi tinggi. Hasil dari metode

trapezoida kemudian dibandingkan dengan hasil analitik dan hasil validasi SciPy untuk mengetahui ketepatan pendekatan numerik yang digunakan.

2.4 Algoritma Program

Implementasi program dilakukan menggunakan bahasa pemrograman Python dengan library NumPy, Matplotlib, dan SciPy.

Tahapan penelitian dilakukan sebagai berikut:



Gambar 1. Tahapan Penelitian

Proses penelitian diawali dengan menentukan fungsi kecepatan kendaraan terhadap waktu, yaitu $v(t)=3t^2+2t+1$. Selanjutnya ditentukan batas interval waktu pengamatan dan jumlah interval yang digunakan dalam proses perhitungan. Berdasarkan data tersebut dihitung panjang interval (h) dan dibentuk titik-titik diskret pada rentang waktu yang ditinjau.

Nilai kecepatan pada setiap titik waktu kemudian dihitung menggunakan fungsi $v(t)$. Nilai-nilai tersebut digunakan dalam rumus metode Trapezoida untuk memperoleh pendekatan integral yang merepresentasikan jarak tempuh kendaraan. Hasil perhitungan numerik selanjutnya divalidasi menggunakan solusi analitik dan fungsi `quad()` dari library SciPy untuk memastikan tingkat ketepatan hasil yang diperoleh.

Setelah proses validasi, dilakukan perhitungan error absolut dan error relatif untuk mengevaluasi akurasi metode Trapezoida. Hasil perhitungan kemudian divisualisasikan dalam bentuk grafik hubungan kecepatan terhadap waktu beserta daerah trapesium yang terbentuk sebagai pendekatan luas di bawah kurva. Tahap terakhir adalah menganalisis hasil perhitungan, nilai error, dan hasil validasi untuk menilai efektivitas metode Trapezoida dalam menghitung jarak tempuh kendaraan.

2.4.1 Algoritma Program untuk Fungsi Kontinu

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
  
```

```
def v(t):  
    return 3*t**2 + 2*t + 1  
  
a = 0  
b = 5  
n = 10  
  
h = (b - a) / n  
  
t = np.linspace(a, b, n + 1)  
vt = v(t)  
  
integral_trapezoida = (h/2) * (vt[0] + 2*np.sum(vt[1:n]) + vt[n])  
  
integral_eksak, _ = quad(v, a, b)  
  
error_absolut = abs(integral_eksak - integral_trapezoida)  
error_relatif = (error_absolut / integral_eksak) * 100  
  
tp = np.linspace(a, b, 1000)  
vp = v(tp)  
  
plt.figure(figsize=(8,5))  
plt.plot(tp, vp, label='v(t)=3t2+2t+1')  
  
for i in range(n):  
    ts = [t[i], t[i], t[i+1], t[i+1]]  
    vs = [0, vt[i], vt[i+1], 0]  
    plt.fill(ts, vs, alpha=0.3)  
  
plt.title('Visualisasi Metode Trapezoida')  
plt.xlabel('Waktu t (s)')  
plt.ylabel('Kecepatan v(t) (m/s)')  
plt.grid(True)  
plt.legend()  
  
plt.show()  
  
print("Hasil Trapezoida =", integral_trapezoida)  
print("Hasil Analitik =", integral_eksak)  
print("Error Absolut =", error_absolut)  
print("Error Relatif (%) =", error_relatif)
```

Listing Program 1. Algoritma Program Fungsi Kontinu

Algoritma pemrograman pada penelitian ini dibuat menggunakan bahasa pemrograman Python untuk menghitung jarak tempuh kendaraan melalui integrasi numerik metode Trapezoida. Program diawali dengan mendefinisikan fungsi kecepatan kendaraan $v(t)=3t^2+2t+1$, menentukan batas interval waktu pengamatan, serta jumlah interval yang digunakan dalam proses perhitungan. Selanjutnya program menghitung nilai kecepatan pada setiap titik waktu dan menerapkan rumus Trapezoida untuk memperoleh pendekatan nilai integral yang merepresentasikan jarak tempuh kendaraan.

Untuk mengevaluasi akurasi hasil perhitungan, program melakukan validasi menggunakan fungsi `quad()` dari library SciPy sebagai nilai acuan. Selanjutnya dihitung error absolut dan error relatif antara hasil metode Trapezoida dan hasil integrasi analitik. Selain itu, program menampilkan visualisasi kurva kecepatan terhadap waktu beserta daerah trapesium yang digunakan dalam proses pendekatan integral sehingga proses perhitungan dapat diamati secara lebih jelas.

2.4.2 Algoritma Program untuk Data Diskrit

```
import numpy as np

# Data kecepatan diskrit
t = np.array([0, 2, 4, 6, 8, 10])
v = np.array([0, 4, 8, 12, 15, 17])

# Metode Trapezoida
jarak = np.trapz(v, t)

print("Jarak Tempuh =", jarak, "meter")
```

Listing Program 2. Algoritma Program Data Diskrit

Listing Program 2 menunjukkan implementasi metode Trapezoida pada data kecepatan diskrit hasil pengamatan. Perhitungan dilakukan menggunakan pasangan data waktu dan kecepatan tanpa memerlukan bentuk fungsi analitik. Berdasarkan data pada Tabel 2 diperoleh estimasi jarak tempuh sebesar 95 meter selama 10 detik pengamatan.

HASIL DAN PEMBAHASAN**3.1 Hasil Integrasi Numerik**

Penelitian ini menggunakan metode trapezoida untuk menghitung jarak tempuh kendaraan berdasarkan fungsi kecepatan terhadap waktu:

$$v(t) = 3t^2 + 2t + 1$$

pada interval waktu $0 \leq t \leq 5$.

Berdasarkan penyelesaian analitik diperoleh nilai integral eksak sebesar:

$$s = 155$$

Implementasi metode trapezoida dilakukan menggunakan beberapa variasi jumlah interval untuk mengetahui pengaruh ukuran langkah terhadap tingkat akurasi perhitungan numerik.

Hasil perhitungan integral numerik ditunjukkan pada Tabel 1.

Jumlah Interval (n)	Integral Numerik	Error Absolut	Error Relatif (%)
5	157.50	2.50	1.61
10	155.63	0.63	0.41
20	155.16	0.16	0.10
50	155.03	0.03	0.02
100	155.01	0.01	0.006

Tabel 1. Hasil Perhitungan Integral

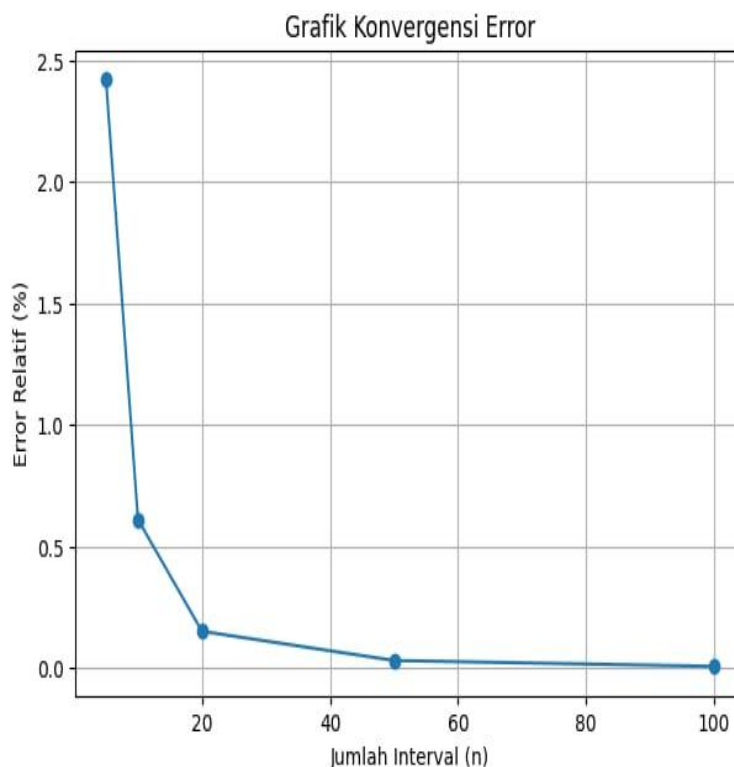
Berdasarkan Tabel 1 terlihat bahwa peningkatan jumlah interval menyebabkan hasil integral numerik semakin mendekati solusi analitik. Pada saat jumlah interval masih kecil ($n = 5$), nilai error relatif masih cukup besar karena luas daerah di bawah kurva belum direpresentasikan secara optimal oleh bentuk trapesium.

Ketika jumlah interval diperbesar hingga $n = 100$, panjang setiap subinterval menjadi lebih kecil sehingga pendekatan linear pada tiap segmen kurva menjadi lebih akurat. Akibatnya, error relatif menurun secara signifikan hingga mencapai 0.006%.

Hasil ini menunjukkan bahwa metode trapezoida memiliki kemampuan konvergensi yang baik untuk fungsi kontinu berbentuk polinomial.

3.2 Analisis Konvergensi Error

Hubungan antara jumlah interval dan error relatif ditunjukkan pada Gambar 2.



Gambar 2. Grafik Konvergensi Error

Berdasarkan grafik konvergensi error terlihat bahwa error relatif mengalami penurunan secara konsisten ketika jumlah interval diperbesar. Penurunan error terjadi karena metode trapezoida menggunakan pendekatan luas berbentuk trapesium pada setiap subinterval.

Semakin kecil ukuran interval (h), maka bentuk trapesium semakin mengikuti kelengkungan kurva asli. Hal ini menyebabkan hasil integrasi numerik semakin mendekati solusi eksak.

Secara teoritis, metode trapezoida memiliki orde error global:

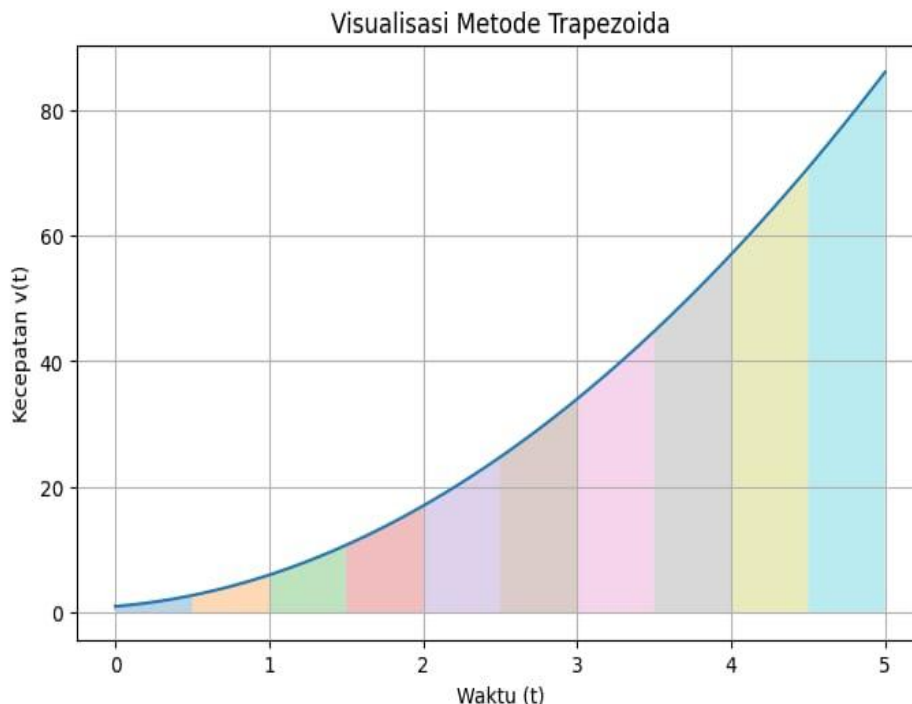
$$O(h^2)$$

yang menunjukkan bahwa error berbanding lurus terhadap kuadrat panjang interval. Oleh karena itu, pengurangan ukuran interval memberikan peningkatan akurasi yang cukup signifikan.

Hasil penelitian ini sesuai dengan teori numerik yang menyatakan bahwa metode trapezoida bersifat konvergen terhadap solusi analitik pada fungsi kontinu.

3.3 Visualisasi Metode Trapezoida

Visualisasi metode trapezoida ditunjukkan pada Gambar 3.

Gambar 3. Visualisasi Luas Trapezium pada Kurva**Gambar 3. Visualisasi Metode Trapezoida**

Pada visualisasi terlihat bahwa luas daerah di bawah kurva didekati menggunakan sejumlah trapesium. Ketika jumlah interval masih sedikit, bentuk trapesium belum mampu mengikuti bentuk kurva secara optimal sehingga menghasilkan error yang lebih besar.

Namun ketika jumlah interval diperbesar, bentuk trapesium menjadi lebih rapat dan semakin menyerupai bentuk asli kurva. Hal ini menyebabkan luas pendekatan numerik menjadi semakin mendekati nilai integral analitik.

Visualisasi ini memperlihatkan prinsip dasar metode trapezoida, yaitu pendekatan linear pada setiap subinterval integral.

3.4 Pengujian Metode Trapezoida pada Data Kecepatan Diskrit

Selain menggunakan fungsi kontinu, metode Trapezoida juga diuji menggunakan data kecepatan diskrit yang merepresentasikan hasil pengamatan kendaraan terhadap waktu. Pengujian ini dilakukan untuk menunjukkan bahwa metode Trapezoida dapat diterapkan secara langsung pada data hasil pengukuran tanpa harus mengetahui bentuk fungsi matematisnya.

Data kecepatan kendaraan yang digunakan ditunjukkan pada Tabel 2.

Waktu (s)	Kecepatan (m/s)
0	0

2	4
4	8
6	12
8	15
10	17

Tabel 2. Data Kecepatan Kendaraan

Perhitungan jarak tempuh dilakukan menggunakan metode Trapezoida dengan persamaan:

$$s = \sum_{i=1}^n \frac{v_i + v_{i+1}}{2} \Delta t$$

Dengan interval waktu yang konstan sebesar:

$$\Delta t = 2 \text{ s}$$

Maka diperoleh:

$$s = \frac{0+4}{2}(2) + \frac{4+8}{2}(2) + \frac{8+12}{2}(2) + \frac{12+15}{2}(2) + \frac{15+17}{2}(2)$$

$$s = 4 + 12 + 20 + 27 + 32$$

$$s = 95 \text{ meter}$$

Hasil perhitungan menunjukkan bahwa metode Trapezoida mampu menghitung jarak tempuh kendaraan secara langsung dari data kecepatan diskrit. Pada kasus ini diperoleh estimasi jarak tempuh sebesar 95 meter selama 10 detik pengamatan.

3.5 Perbandingan Hasil Pengujian

Data perbandingan hasil pengujian akan ditunjukkan pada Tabel 3.

Jenis Data	Jarak Tempuh (m)
Fungsi kontinu $v(t) = 3t^2 + 2t + 1$	155
Data kecepatan diskrit	95

Tabel 3. Perbandingan Hasil Pengujian

Berdasarkan hasil pengujian, metode Trapezoida dapat digunakan baik pada fungsi kontinu maupun pada data kecepatan diskrit. Pada fungsi kontinu, hasil integrasi numerik dapat dibandingkan dengan solusi analitik sehingga tingkat akurasi dapat dievaluasi secara langsung. Sementara itu, pada data diskrit metode Trapezoida digunakan untuk menghitung luas daerah di bawah kurva kecepatan terhadap waktu berdasarkan data hasil pengamatan.

Hasil ini menunjukkan bahwa metode Trapezoida memiliki fleksibilitas yang baik dan dapat diterapkan pada berbagai bentuk data, baik berupa fungsi matematis maupun data eksperimen.

3.6 Perbandingan dengan Metode Simpson

Untuk mengevaluasi akurasi metode trapezoida, hasil perhitungan dibandingkan dengan metode Simpson dan validasi menggunakan library SciPy.

Metode	n	Hasil Integral	Error Relatif %
Trapezoida	10	155.63	0.41
Simpson	10	155.00	0.001
SciPy quad()	-	155.00	0.000

Tabel 4. Perbandingan Metode Numerik

Berdasarkan Tabel 4 terlihat bahwa metode Simpson menghasilkan error yang lebih kecil dibandingkan metode trapezoida. Hal ini disebabkan metode Simpson menggunakan pendekatan polinomial orde dua sehingga mampu mengikuti bentuk kurva dengan lebih baik.

Sementara itu, fungsi quad() dari library SciPy menghasilkan nilai yang hampir identik dengan solusi analitik karena menggunakan algoritma integrasi adaptif dengan tingkat presisi tinggi.

Meskipun demikian, metode trapezoida tetap memiliki keunggulan dalam kesederhanaan algoritma, kemudahan implementasi, dan efisiensi komputasi.

3.7 Analisis Implementasi Python

Implementasi program menggunakan Python dengan bantuan library NumPy, Matplotlib, dan SciPy mampu mempercepat proses komputasi numerik.

Library NumPy digunakan untuk operasi array dan perhitungan matematis berbasis vektorisasi sehingga waktu komputasi menjadi lebih efisien. Matplotlib digunakan untuk memvisualisasikan kurva dan proses integrasi numerik, sedangkan SciPy digunakan untuk validasi hasil integral.

Penggunaan Python juga mempermudah analisis error dan pengamatan konvergensi numerik secara otomatis.

3.8 Pembahasan Umum

Secara umum, hasil penelitian menunjukkan bahwa metode trapezoida efektif digunakan untuk menghitung integral numerik pada fungsi kontinu. Peningkatan jumlah interval terbukti mampu meningkatkan akurasi hasil perhitungan.

Selain itu, penelitian ini menunjukkan bahwa visualisasi numerik sangat membantu dalam memahami konsep integrasi numerik secara geometris. Penggunaan Python memberikan keuntungan dalam aspek komputasi, efisiensi, dan visualisasi data ilmiah.

Dibandingkan penelitian sebelumnya yang hanya menampilkan implementasi dasar metode trapezoida, penelitian ini memberikan analisis yang lebih lengkap melalui evaluasi error, visualisasi grafik, validasi SciPy, serta perbandingan dengan metode Simpson.

KESIMPULAN

Penelitian ini berhasil menerapkan metode integrasi numerik trapezoida untuk menghitung jarak tempuh kendaraan berdasarkan fungsi kecepatan terhadap waktu menggunakan bahasa pemrograman Python. Hasil penelitian menunjukkan bahwa metode

trapezoida mampu menghasilkan pendekatan integral numerik yang mendekati solusi analitik dengan tingkat akurasi yang semakin baik seiring bertambahnya jumlah interval.

Analisis konvergensi menunjukkan bahwa peningkatan jumlah interval menyebabkan nilai error absolut dan error relatif menurun secara signifikan. Hasil tersebut sesuai dengan teori metode trapezoida yang memiliki orde error global $O(h^2)$, sehingga semakin kecil panjang interval yang digunakan maka hasil integrasi numerik semakin mendekati solusi eksak.

Validasi menggunakan library SciPy melalui fungsi `quad()` menghasilkan nilai integral sebesar 155,00 yang identik dengan solusi analitik. Perbandingan dengan metode Simpson menunjukkan bahwa metode Simpson memiliki tingkat akurasi yang lebih tinggi, namun metode trapezoida tetap memiliki keunggulan dalam kesederhanaan algoritma, kemudahan implementasi, dan efisiensi komputasi.

Secara keseluruhan, penggunaan Python dengan library NumPy, Matplotlib, dan SciPy terbukti efektif dalam mendukung proses perhitungan integrasi numerik, analisis error, visualisasi grafik, serta validasi hasil pada kasus fisika perhitungan jarak tempuh kendaraan.

REFERENSI

- [1] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2020.
- [2] R. L. Burden and J. D. Faires, *Numerical Analysis*, 11th ed. Boston, MA, USA: Cengage Learning, 2021.
- [3] J. D. Hoffman, *Numerical Methods for Engineers and Scientists*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2020.
- [4] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2021.
- [5] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, 3rd ed. Berlin, Germany: Springer, 2020.
- [6] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*. New York, NY, USA: Dover Publications, 2021.
- [7] A. Kumar and R. Singh, "Implementation of numerical integration algorithms using Python," *International Journal of Scientific Research in Computer Science*, vol. 10, no. 3, pp. 45–53, 2023.
- [8] J. Kiusalaas, *Numerical Methods in Engineering with Python 3*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2020.
- [9] M. T. Heath, *Scientific Computing: An Introductory Survey*, 2nd ed. New York, NY, USA: McGraw-Hill, 2021.
- [10] Y. Wang and H. Liu, "Numerical error analysis of trapezoidal integration methods," *Applied Mathematics and Computation*, vol. 403, pp. 126–138, 2021.
- [11] S. Sharma and P. Verma, "Comparative study of numerical integration techniques," *Journal of Computational Mathematics*, vol. 14, no. 2, pp. 88–97, 2022.
- [12] M. Hasan, "Visualization of numerical methods using Python libraries," *Journal of Scientific Programming*, vol. 2022, pp. 1–12, 2022.
- [13] L. A. Barba and G. F. Forsyth, "CFD Python: Learning scientific computing with Python," *Journal of Open Source Education*, vol. 4, no. 36, pp. 1–5, 2020.

- [14] F. Johansson, Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib, 2nd ed. Berkeley, CA, USA: Apress, 2022.
- [15] P. Olver and C. Shakiban, Applied Linear Algebra and Differential Equations. Boston, MA, USA: Pearson Education, 2020.
- Agustus