Untitled0.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:00 PM

Comment Share M

+ Code + Text

RAM Disk Gemini

```python
# Mengimport Library
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as integrate
from sympy import symbols, cos, integrate as sympy_integrate

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

# Batas integrasi
a = 1.0  # Batas bawah
b = 5.0  # Batas atas
n = 10   # Jumlah grid, harus ganjil untuk metode Simpson

# METODE EKSAK
# Menggunakan Sympy untuk menghitung integral eksak secara simbolik
x_symbol = symbols('x')
f_symbolic = x_symbol**(-3) + cos(x_symbol)

# Integral eksak antara batas a dan b
integral_eksak = sympy_integrate(f_symbolic, (x_symbol, a, b))

# SIMPSON'S RULE
# Jika n % 2 == 0, maka jumlah n harus ganjil, sehingga ditambah 1
if n % 2 == 0:
    n += 1  # Jika n genap, tambahkan 1 agar menjadi ganjil

# Membuat grid atau titik-titik x dengan jarak yang sama antara a dan b
```

Connected to Python 3 Google Compute Engine backend

Untitled0.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:00 PM

Comment Share M

+ Code + Text

RAM Disk Gemini

```python
# Membuat grid atau titik-titik x dengan jarak yang sama antara a dan b
x = np.linspace(a, b, n)
dx = (x[-1] - x[0]) / (n - 1)  # Menghitung lebar tiap grid (dx)

# Menghitung integral menggunakan metode Simpson
# Mulai dengan menjumlahkan fungsi di titik pertama dan terakhir (f(a) dan f(b))
hasil_simpson = func(x[0]) + func(x[-1])

# Menghitung bagian untuk indeks ganjil
# Indeks ganjil dimulai dari 1 hingga n-2, dikalikan 4
for i in range(1, n-1, 2):
    hasil_simpson += 4 * func(x[i])  # Penjumlahan 4 * f(x_i)

# Menghitung bagian untuk indeks genap
# Indeks genap dimulai dari 2 hingga n-3, dikalikan 2
for i in range(2, n-2, 2):
    hasil_simpson += 2 * func(x[i])  # Penjumlahan 2 * f(x_i)

# Kalikan hasil akhir dengan dx / 3 sesuai dengan aturan Simpson
hasil_simpson = hasil_simpson * dx / 3

# METODE TRAPEZOID
# Membuat grid yang lebih halus untuk metode trapezoid
x_trapz = np.linspace(a, b, n)
y_trapz = func(x_trapz)

# Menggunakan scipy.integrate.trapz untuk menghitung dengan metode trapezoid
hasil_trapezoid = integrate.trapz(y_trapz, x_trapz)

# Visualisasi grafik Eksak
xp = np.linspace(a, b, 1000)
```

Connected to Python 3 Google Compute Engine backend

Untitled0.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:00 PM

Comment Share M

+ Code + Text

RAM Disk Gemini

```python
# Visualisasi grafik Eksak
xp = np.linspace(a, b, 1000)
plt.figure(figsize=(12, 6))

# Grafik untuk metode eksak
plt.subplot(1, 3, 1)
plt.plot(xp, func(xp), label="Grafik Eksak", color='blue')
plt.fill_between(xp, func(xp), color='lightblue', alpha=0.5)
plt.title('Grafik Eksak')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.legend()

# Grafik untuk metode Simpson
plt.subplot(1, 3, 2)
plt.plot(xp, func(xp), label="Grafik Simpson", color='green')
for i in range(n):
    plt.bar(x[i], func(x[i]), align='edge', width=dx, color='red', edgecolor='black')
plt.title('Grafik Simpson')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.legend()

# Grafik untuk metode Trapezoid
plt.subplot(1, 3, 3)
plt.plot(xp, func(xp), label="Grafik Trapezoid", color='purple')
plt.fill_between(x_trapz, y_trapz, color='lightgreen', alpha=0.5)
plt.plot(x_trapz, y_trapz, 'ro--', label='Titik Trapezoid')
plt.title('Grafik Trapezoid')
plt.xlabel('x')
plt.ylabel('f(x)')
```

Connected to Python 3 Google Compute Engine backend

Untitled0.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:00 PM

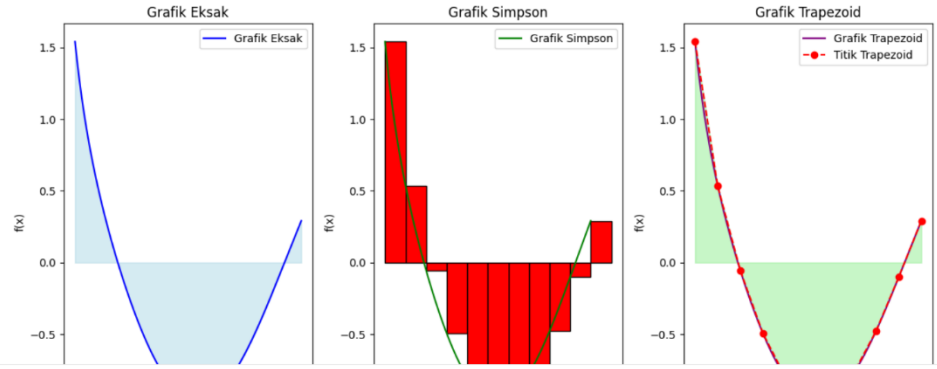
Comment Share M

+ Code + Text

RAM Disk Gemini

```
# Cetak hasil integral dengan urutan: metode eksak, Simpson, dan Trapezoid
print(f"Hasil integral eksak: {integral_eksak.evalf()}")
print(f"Hasil integral menggunakan metode Simpson: {hasil_simpson}")
print(f"Hasil integral menggunakan metode Trapezoid: {hasil_trapezoid}")
```

```
<ipython-input-2-0d46e042030e>:57: DeprecationWarning: 'scipy.integrate.trapz' is deprecated in favour of 'scipy.integrate.trapezoid' and will be removed in SciPy 1.14.0
  hasil_trapezoid = integrate.trapz(y_trapz, x_trapz)
```



Connected to Python 3 Google Compute Engine backend

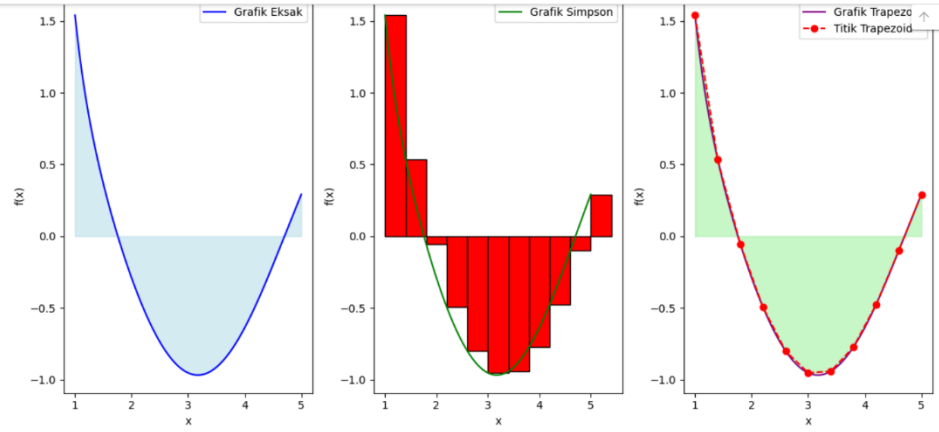
Untitled0.ipynb

File Edit View Insert Runtime Tools Help Last saved at 6:00 PM

Comment Share M

+ Code + Text

RAM Disk Gemini




```
Hasil integral eksak: -1.32039525947103
Hasil integral menggunakan metode Simpson: -1.3155721992540337
Hasil integral menggunakan metode Trapezoid: -1.2582519333988034
```

Connected to Python 3 Google Compute Engine backend