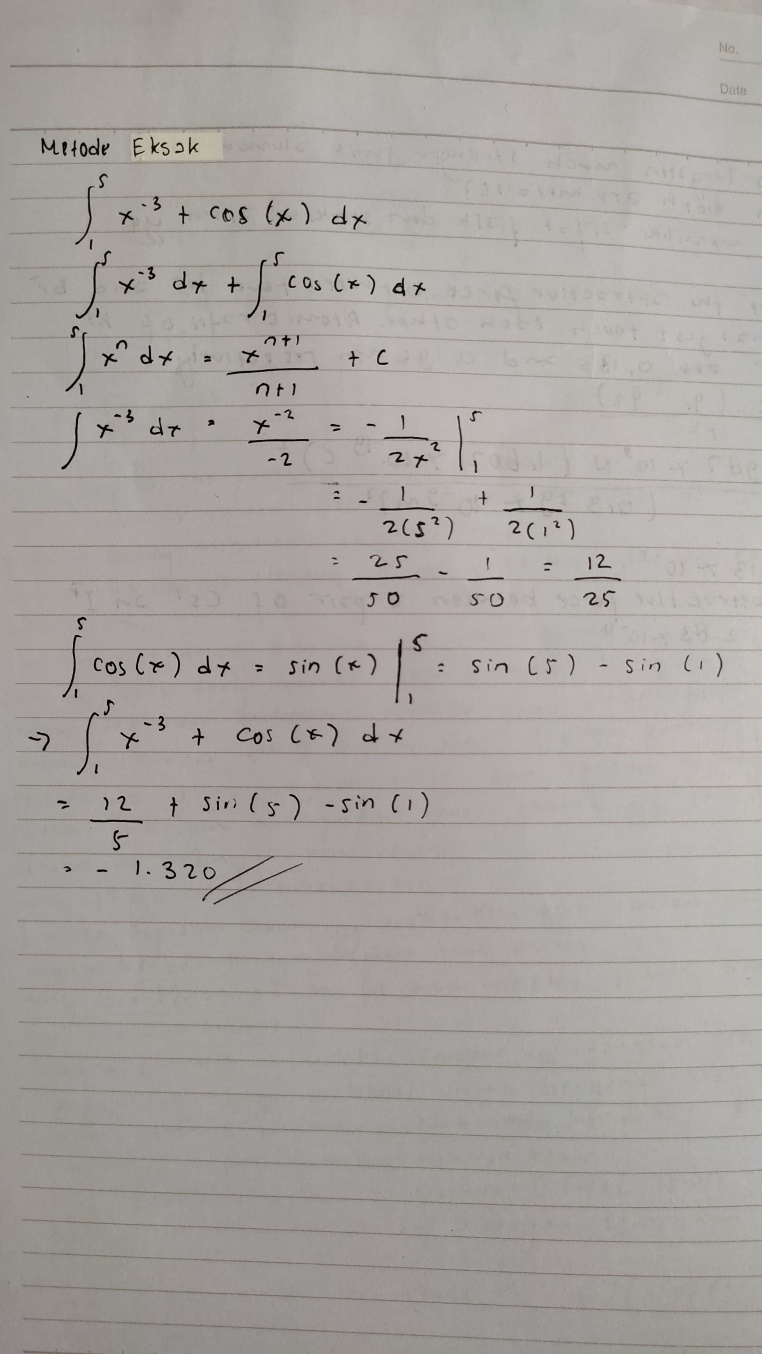
**Praktikum Fisika Komputasi**

**Modul 4**

**Septian Tri Laksono**

**1227030032**

**Metode Eksak**

****

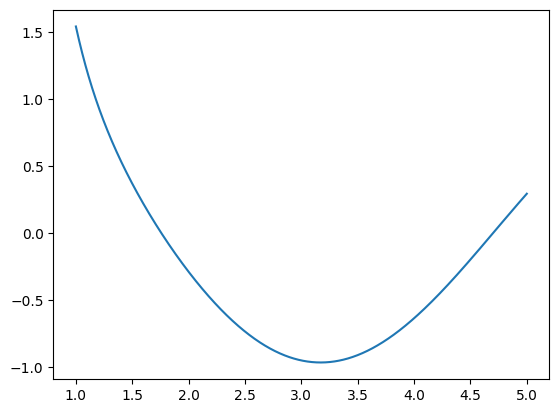
**Metode Trapezoid**

|  |
| --- |
| # import Library  import numpy as np  import matplotlib.pyplot as plt |

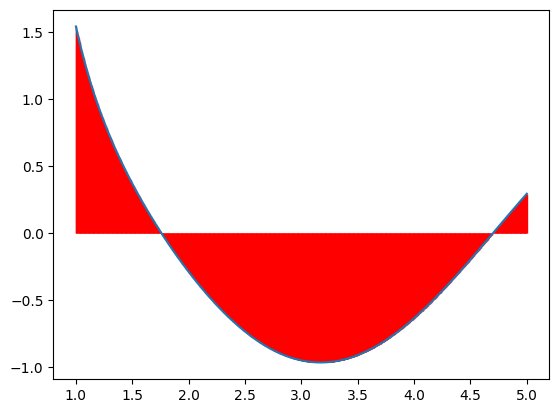
|  |
| --- |
| #Integral  def func(x):                  #Nama fungsi      return (x\*\*-3) + np.cos(x)   #fungsi yg diintegralkan  a = 1.0                       #batas bawah  b = 5.0                      #batas atas  # Menghitung integral  integral, error = quad(func, a, b)  print(f"Nilai integral: {integral}") |

|  |
| --- |
| #metode trapezoid  n = 1000  #jumlah grid  dx = (b-a)/(n-1)  x = np.linspace(a,b,n)  sigma = 0  for i in range (1, n-1):      sigma += func(x[i])  hasil = 0.5\*dx\*(func(x[0])+2\*sigma+func(x[-1]))  print (hasil) |

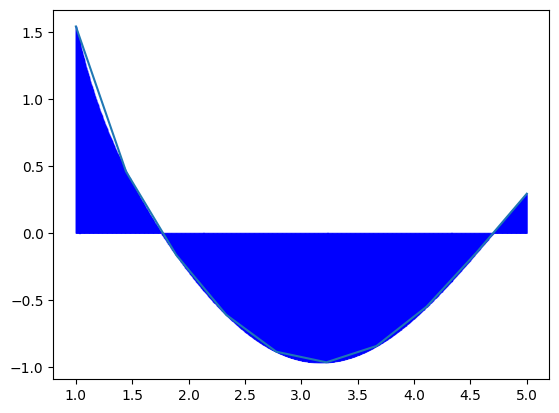
|  |
| --- |
| xp =np.linspace(a,b,1000)  plt.plot(xp,func(xp))  plt.show() |



|  |
| --- |
| xp =np.linspace(a,b,1000)  plt.plot(xp,func(xp))  for i in range (n):      plt.bar(x[i],func(x[i]), align = 'edge',width = 0.000001, edgecolor='red')  plt.show() |

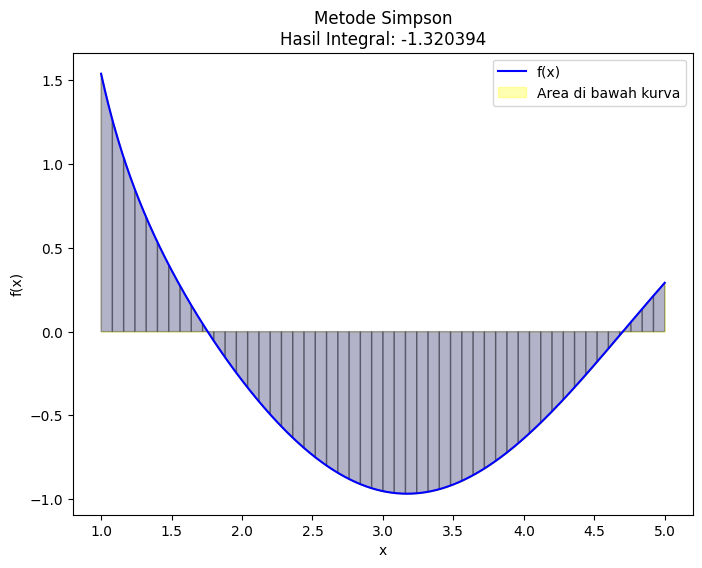


|  |
| --- |
| xp =np.linspace(a,b,10)  plt.plot(xp,func(xp))  for i in range (n):       plt.bar(x[i],func(x[i]), align = 'edge',width = 0.000001, edgecolor='blue')  plt.fill\_between(x,func(x),color= 'blue', alpha=0.5)  plt.show() |



**Metode Simpson 1/3**

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  from scipy.integrate import simps  # Definisikan fungsi  def f(x):      return x\*\*(-3) + np.cos(x)  # Batas integral dan jumlah partisi  a = 1  b = 5  n = 1000  # Pastikan n genap untuk metode Simpson  # Buat array x dan y  x = np.linspace(a, b, n+1)  y = f(x)  # Implementasi metode Simpson  integral\_simpson = simps(y, x)  # Plot untuk metode Simpson  x\_fine = np.linspace(a, b, 100)  y\_fine = f(x\_fine)  plt.figure(figsize=(8, 6))  plt.plot(x\_fine, y\_fine, label="f(x)", color="blue")  plt.fill\_between(x, 0, y, color='yellow', alpha=0.3, label='Area di bawah kurva')  # Plot area untuk metode Simpson  for i in range(0, n, 2):      plt.fill([x[i], x[i], x[i+2], x[i+2]], [0, y[i], y[i+2], 0], color='blue', edgecolor='black', alpha=0.3)  plt.title(f"Metode Simpson\nHasil Integral: {integral\_simpson:.6f}")  plt.xlabel('x')  plt.ylabel('f(x)')  plt.legend()  # Tampilkan grafik  plt.show() |



Dalam segi akurasi, metode eksak memberikan hasil yang paling akurat karena menghasilkan solusi analitis. Sementara itu, metode Simpson lebih akurat dibandingkan metode trapezoid dalam pendekatan numerik, karena pendekatan dengan parabola lebih sesuai dengan bentuk kurva. Metode trapezoid lebih mudah diterapkan, tetapi kurang akurat untuk fungsi yang lebih kompleks atau dengan perubahan kurva yang tajam.

Metode eksak bisa menjadi lebih rumit secara matematis, terutama jika fungsi yang akan diintegralkan cukup sulit. Metode trapezoid lebih sederhana dan mudah diaplikasikan dibandingkan metode Simpson, meskipun akurasinya lebih rendah. Metode Simpson memerlukan perhitungan yang lebih rumit daripada metode trapezoid, tetapi menawarkan hasil yang lebih akurat.

Untuk fungsi yang sederhana atau dapat diintegrasikan secara analitis, metode eksak adalah pilihan terbaik. Namun, jika fungsi tidak dapat diintegralkan secara analitis dan pendekatan numerik dibutuhkan, metode Simpson cenderung lebih efektif. Metode trapezoid adalah alternatif yang cepat dengan perhitungan sederhana, tetapi jika akurasi lebih diutamakan, metode Simpson lebih disarankan. Secara keseluruhan, metode Simpson biasanya lebih efektif daripada metode trapezoid dalam pendekatan numerik, sementara metode eksak tetap menjadi pilihan terbaik jika memungkinkan.