Reza Farel Ramdhani
NIM.1227030028

PERHITUNGAN EKSAK

Pertanyaan :

$$\int_1^5 x^{-3} + \cos(x)\, dx$$

Solusi:

$$\int_1^5 x^{-3} + \cos(x)\, dx = \int_1^5 x^{-3}\, dx + \int_1^5 \cos(x)\, dx$$

$$\int_1^5 x^{-3} + \cos(x)\, dx = [-\tfrac{1}{2}x^{-2}]_1^5 + [\sin(x)]_1^5$$

$$\int_1^5 x^{-3} + \cos(x)\, dx = \left\{\left(-\frac{1}{50}\right) - \left(\frac{1}{2}\right)\right\} + \{(\sin(5)) - (\sin(1))\}$$

$$\int_1^5 x^{-3} + \cos(x)\, dx = -0{,}02 - 0{,}5 + (-0{,}97892427) - 0{,}8414709$$

$$\int_1^5 x^{-3} + \cos(x)\, dx = -1{,}3203952593882757$$

PERHITUNGAN METODE TRAPEZOID

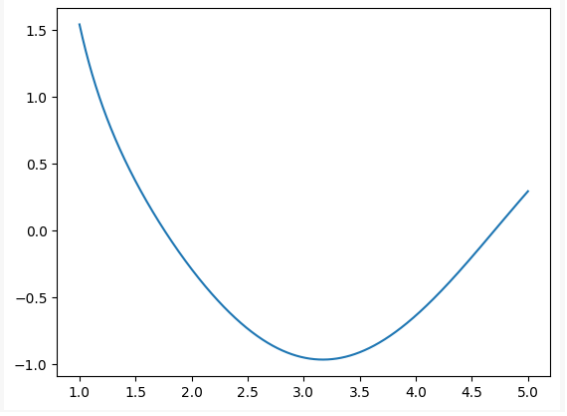
```
#METODE TRAPEZOID
#Reza Farel Ramdhani
#NIM.1227030028

#Mengimport Library
import numpy as np
import matplotlib.pyplot as plt

#Integral
def func(x):                  #Nama fungsi
    return (x**(-3))+np.cos(x) #Fungsi yang akan diintegralkan
a = 1.0                       #Batas bawah
b = 5.0                     #Batas atas

#Metode Trapezoid
n = 100
dx = (b-a)/(n-1)
x = np.linspace(a,b,n)

sigma = 0
for i in range(1, n-1):
    sigma += func(x[i])
```

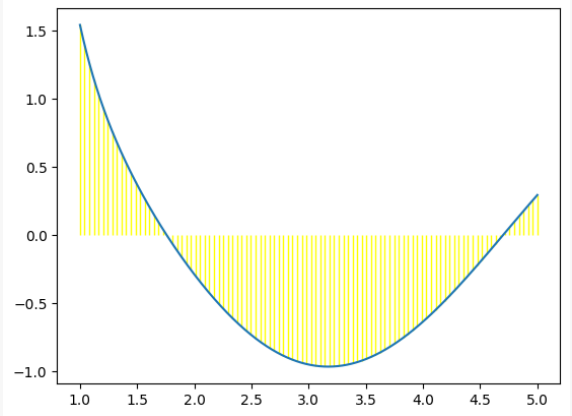
Reza Farel Ramdhani
NIM.1227030028

```
hasil = 0.5*dx*(func(x[i])+2*sigma+func(x[-1]))

print(hasil)

#Grafik 1
xp =np.linspace(a,b,1000)
plt.plot(xp,func(xp))
plt.show()

#Grafik 2
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='yellow')

plt.show()

#Grafik 3
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='yellow')

plt.fill_between(x,func(x),color='red', alpha=0.6)

plt.show()
```

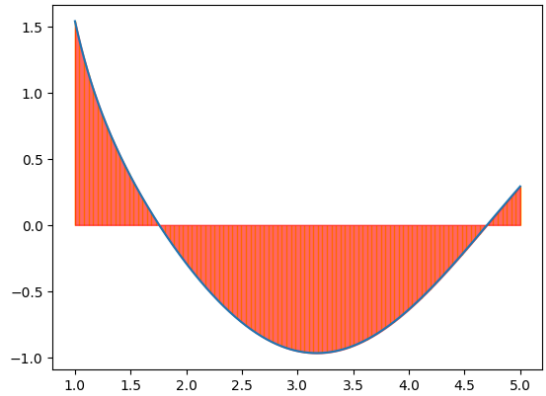
Hasil perhitungan:

```
-1.345751329330685
```





Reza Farel Ramdhani
NIM.1227030028



PERHITUNGAN METODE SIMPSON 1/3

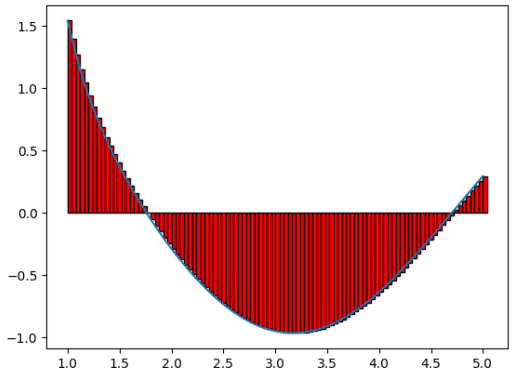
```
#Metode Simpson 1/3
#Reza Farel Ramdhani
#NIM.1227030028

#Mengimport Library
import numpy as np
import matplotlib.pyplot as plt

#Fungsi Integral
def func(x):                    #Nama fungsi
    return (x**(-3))+np.cos(x)  #Fungsi yang akan diintegralkan
a = 1.0                         #Batas bawah
b = 5.0                        #Batas atas
n = 100                         #Jumlah grid, harus ganjil untuk metode
simpson
#Simpson's Rule
if n % 2 == 0:  #Jika n nya dibagi 2 dan hasilnya genap, maka tidak
memenuhi aturan Simpson
    n += 1      #Jika n genap, tambah 1 agar menjadi ganjil

x = np.linspace(a,b,n)
dx = (x[-1]-x[0])/(n-1)

#Menghitung integral menggunakan metode Simpson
hasil = func(x[0])+func(x[-1])    #Tambah f(a) dan f(b)

for i in range(1,n-1,2):
    hasil += 4*func(x[i])         #Untuk indeks ganjil

for i in range(2,n-2,2):
    hasil += 2*func(x[i])         #Untuk indeks genap

hasil*= dx/3                      #Faktor dx/3
```

Reza Farel Ramdhani
NIM.1227030028

```
#Visualisasi grafik dan bar
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=dx, color='red',
edgecolor='black')
plt.show()
print(hasil)
```

Hasil perhitungan:

```
-1.3203944385483368
```



ANALISIS PERHITUNGAN

Berdasarkan percobaan yang telah dilakukan, didapatkan data hasil perhitungann secara dengan metode eksak sebesar $-1{,}3203952593882757$, metode trapezoid sebesar -1.345751329330685, dan metode simpson 1/3 sebesar -1.3203944385483368. Berdasarkan data tersebut, diketahui bahwa metode simpson 1/3 merupakan metode yang memiliki hasil yang lebih akurat dengan perhitungan eksak disbandingkan dengan metode trapezoid.

Metode trapezoid menggunakan prinsip pendekatan daerah bawah grafik fungsi (x) sebagai trapesium dengan lebar yang sama, kemudian menghitung luas setiap trapesium dan menjumlahkannya untuk mendapatkan pendekatan nilai integral. Metode simpson 1/3 menggunakan persamaan polinomial orde dua, serta metode ini merupakan perluasan dari aturan trapesium. Sehingga berdasarkan hal tersebut, dapat disimpulkan bahwa metode simpson 1/3 lebih baik digunakan dibandingkan dengan metode trapezoid untuk perhitungan yang memerlukan akurasi tinggi.