Laporan Praktikum Fisika Komputasi

Tugas 3 : Senin, 07 Oktober 2024

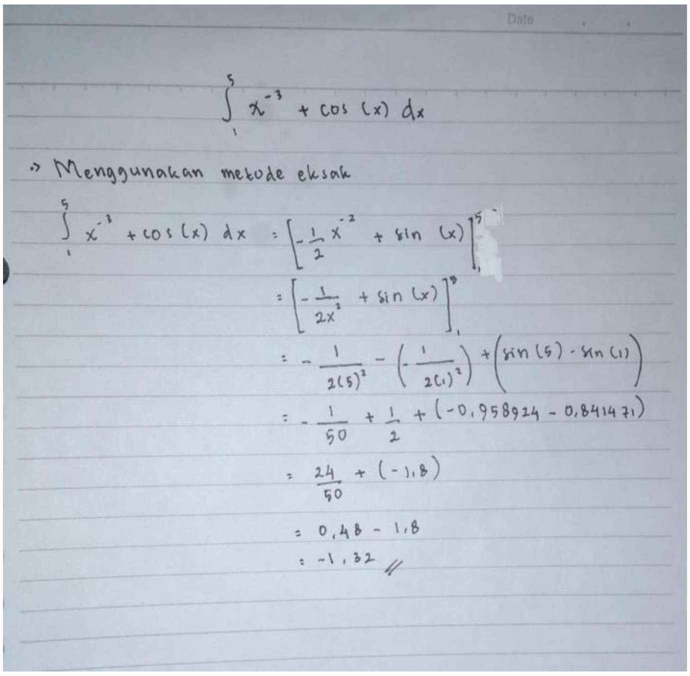
Disusun oleh : Najlah Rupaidah

NIM : 1227030025

1. Praktikum ini berkenaan tentang penyelesaian soal integral dengan menggunakan kode program pada *software* colab.google. Dalam menyelesaikan soal integral ini, dapat dilakukan dalam beberapa metode, diantaranya yaitu:

a) Metode Eksak

Metode ini menggunakan perhitungan manual dalam menyelesaikan persoalan integral.



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

→ Menggunakan metode eksak

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

$$= \left[-\frac{1}{2x^2} + \sin(x)\right]_1^5$$

$$= -\frac{1}{2(5)^2} - \left(-\frac{1}{2(1)^2}\right) + \left(\sin(5) - \sin(1)\right)$$

$$= -\frac{1}{50} + \frac{1}{2} + (-0{,}958924 - 0{,}841471)$$

$$= \frac{24}{50} + (-1{,}8)$$

$$= 0{,}48 - 1{,}8$$

$$= -1{,}32$$

Setelah di hitung secara manual, di dapatkan hasil integral dari persamaannya sebesar -1,32.

b) Metode Trapezoid

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

Pertama, mengimpor dua library, yaitu Numpy dan Matplotlib. Numpy digunakan untuk membantu melakukan berbagai perhitungan matematika. Sementara itu, Matplotlib, khususnya modul pyplot, digunakan untuk membuat berbagai grafik dan visualisasi data.

```
#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
```

Kode program ini mendefinisikan sebuah fungsi matematika bernama func (x).

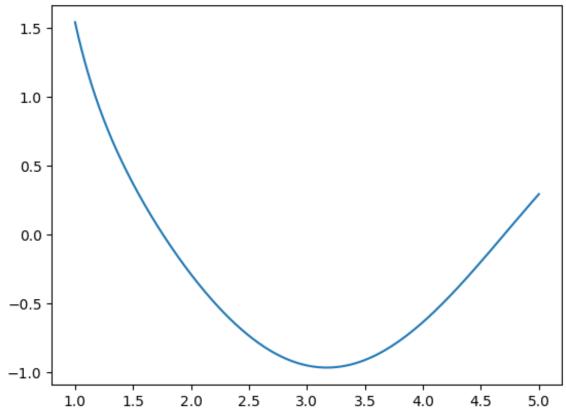
Lalu, memasukkan fungsi yang akan dicari nilai integralnya. Kode program juga menetapkan dua variabel, a dan b, yang masing-masing bernilai 1.0 dan 5.0. Nilai ini akan berfungsi sebagai batas bawah dan batas atas untuk perhitungan integral yang nantinya dilakukan terhadap fungsi tersebut. Namun, dalam potongan kode ini, integralnya sendiri belum dihitung, hanya didefinisikan batas-batasnya.

```
#Metode Trapezoid
n = 200                #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)
```

Selanjutnya pada bagian ini, jumlah grid (n) diatur menjadi 200. Untuk grid ini dapat disesuaikan dengan keinginan masing-masing. Kemudian, nilai fungsi dihitung di setiap titik kecuali titik awal dan akhir, dan hasilnya dijumlahkan dalam variabel **sigma**. Setelah itu, integral dihitung menggunakan formula trapezoid, dan hasil akhir integral kemudian dicetak. Setelah di run, pada bagian bawah kode program akan muncul hasilnya sebesar -1.3202338088833576.

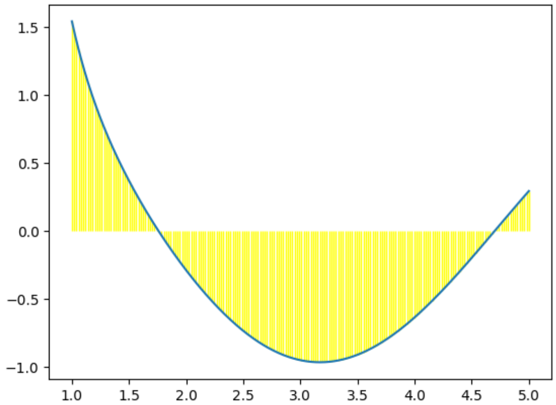
```
xp =np.linspace(a,b,1000)
plt.plot(xp,func(xp))
plt.show ()
```

Bagian ini, kode tersebut fungsinya untuk membuat dan menampilkan grafik dari fungsi matematisnya. Pertama, **xp = np.linspace(a, b, 1000)** menghasilkan 1000 titik antara **a** dan **b**. Lalu, **plt.plot(xp, func(xp))** menggambar kurva fungsi **func** terhadap nilai **xp**. Terakhir, **plt.show** menampilkan grafik dibawah ini:



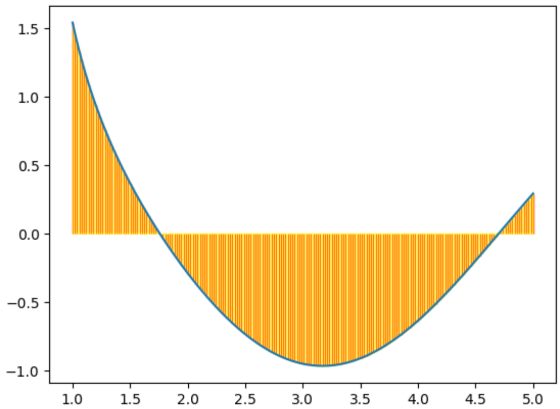
```
xp =np.linspace(a,b,1000)
plt.plot(xp,func(xp))

for i in range (n):
     plt.bar(x[i],func(x[i]), align = 'edge',width = 0.000001,
edgecolor='yellow')

plt.show()
```

Kode ini pun membuat grafik fungsi matematika. Pertama, ia menghasilkan 1000 titik x antara **a** dan **b** menggunakan **np.linspace**. Lalu, grafik fungsi **func** tersebut digambar dengan **plt.plot**. Akhirnya, grafik ditampilkan dengan **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.fill_between(x,func(x),color= 'yellow', alpha=0.5)

plt.show()
```

Terakhir, yaitu membuat grafik kembali, hanya saja untuk kode program ini terdapat penambahan yaitu ia menambahkan batang merah di titik tertentu dengan lebar sangat kecil, sebelum menampilkan grafik menggunakan **plt.show()**.



c) Metode Simpson 1/3

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

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

# Batas integrasi
a = 1.0  # Batas bawah
b = 5.0  # Batas atas
n = 200  # Jumlah grid, harus ganjil untuk metode Simpson
```

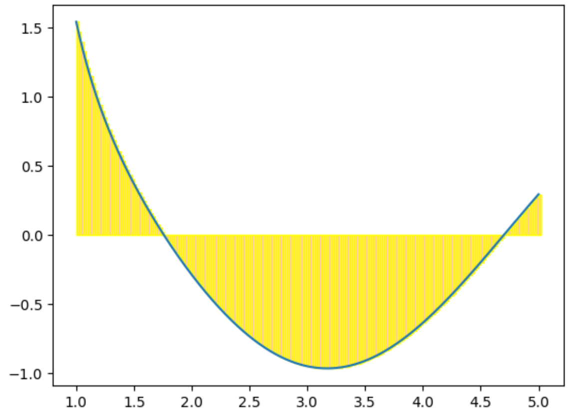
Pertama yaitu mengimpor *library* **numpy** dan **matplotlib.pyplot** untuk perhitungan numerik dan visualisasi. Lalu, sebuah fungsi **func(x)** didefinisikan, dengan memasukkan fungsi yang akan diintegralkan. Selanjutnya, batas integrasi didefinisikan dengan nilai bawah a = 1.0, nilai atas b = 5.0, dan jumlah grid n = 200, yang perlu ganjil untuk metode Simpson.

```
# Simpson's Rule
if n % 2 == 0:
    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

# 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='pink', edgecolor='yellow')
```

```
plt.show()
print(hasil)
```

Selanjutnya, terdapat peraturan Simpson, dimana jika jumlah pembagian interval **n** adalah genap, **n** ditambah 1 agar menjadi ganjil, karena metode Simpson memerlukan **n** ganjil. Integral dihitung dengan menjumlahkan nilai fungsi di titik-titik awal dan akhir f(a) dan f(b), kemudian menambahkan hasil dari titik-titik ganjil dikalikan 4 dan titik-titik genap dikalikan 2, sesuai aturan Simpson. Hasilnya dikalikan dengan dx/3 untuk mendapatkan nilai integral.

Akhirnya, grafik fungsi divisualisasikan dengan batang-batang yang menggambarkan area di bawah kurva, dan hasil integral ditampilkan sebesar -1.3203952078478876



2. Pada **metode eksak**, fungsi dihitung dan dievaluasi pada batas-batas yang ditentukan secara manual. Hasil yang diperoleh adalah sekitar −1.32. Sementara itu, **metode trapezoid** menggunakan pendekatan numerik. Dengan membagi area di bawah kurva menjadi beberapa trapesium/trapezoid, metode ini menghitung luas masing-masing trapezoid dan menjumlahkannya untuk mendapatkan hasilnya, pada soal fungsi ini hasilnya yaitu sebesar -1.3203952078478876. Di sisi lain, **metode Simpson 1/3** merupakan perluasan dari aturan trapezoid, karena menggunakan polinomial kuadratik untuk mendekati fungsi dalam interval yang lebih kecil. Metode ini biasanya menghasilkan hasil yang lebih mendekati nilai eksak, untuk hasil dari fungsi yang ditanyakan menggunakan metode ini yaitu sebesar -1.3203952078478876. Secara keseluruhan, ketiga metode ini mengonfirmasi bahwa integral dari fungsi yang dianalisis, di antara batas 1 dan 5, adalah sekitar −1.32, -1.3203952078478876, dan -1.3203952078478876. Masing-masing metode memiliki cara kerja dan tingkat akurasi yang berbeda, namun semuanya dapat digunakan untuk menentukan nilai integral.

3. Dalam hal pemahaman, metode eksak adalah yang terbaik jika kita bisa menemukan antiderivatifnya, karena kecenderungan sering menggunakan metode ini dalam sehari-hari. Namun terkadang pada metode eksak ini membutuhkan ketelitian yang tinggi karena bagaimanapun perhitungan secara manual sangat riskan salah dalam menghitung. Tapi, jika fungsi sulit untuk diintegrasikan, metode Simpson 1/3 biasanya lebih efektif dibandingkan metode trapezoid karena hasilnya lebih akurat. Pilihan metode juga tergantung pada bentuk fungsi dan seberapa akurat kita membutuhkannya, jadi metode Simpson 1/3 merupakan metode yang lebih efektif untuk perhitungan numerik yang akurat.