PRAKTIKUM FISIKA KOMPUTASI

MODUL 4 INTEGRAL METODE NUMERIK

Nama:

AZHAR RIZKY AULIA (1227030008)

$$\int_{1}^{5} x^{-3} + \cos(x)\, \mathrm{dx}$$

Metode Eksak

Metode Eksak

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

Penyelesaian:

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

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

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

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

$$= \frac{-1}{50} + \frac{1}{2} + (-0{,}9589 - 0{,}8415)$$

$$= \frac{24}{50} + (-1{,}8004)$$

$$= 0{,}48 - 1{,}8004$$

$$= -1{,}3204$$

Hasil yang didapatkan melalui metode eksak ini adalah – 1.3204, dan cara mendapatkan hasil tersebut adalah dengan menggunakan perhitungan secara langsung dengan menyelesaikan masing-masing integral dari soal tersebut, yaitu pertama adalah menghitung integral dari x^{-3} dimana didapatkan hasil $\frac{1}{-2\,x^{2}}$ kemudian menghitung integral dari cos(x) dimana didapatkan hasilnya yaitu sin(x). Setelah didapatkan hasil perhitungan kedua integral tersebut selanjutkan hasil tersebut digabungkan, dan dimasukkan batas atas dan batas bawahnya, dan dilakukan perhitungan sehingga didapatkan hasil -1.3204.

Metode Trapezoid

```
import numpy as np
import matplotlib.pyplot as plt
```

Kode ini berfungsi untuk mengimport library numpy dan matplotlib yang akan digunakan pada kode program. Numpy untuk perhitungan numerik, dan matplotlib untuk membuat plot grafik.

```
def func(x):
    return x**-3 + np.cos(x)

a = 1.0
b = 5.0
```

Kode ini berfungsi untuk mendefinisikan sebuah fungsi yang akan diintegralkan, dimana fungsinya adalah $f(x) = x^{-3} + \cos(x)$ dan mendefisinikan batas atas dan batas bawah integral, dimana batas atasnya adalah 5 dan batas bawahnya adalah 1.

```
n = 150
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)
```

Kode ini berisi pendefinisian jumlah garis yang akan digunakan untuk membagi area dibawah kurva, dan digunakan 150 garis. Kemudian terdapat variabel dx dengan berisi perhitungan yang berfungsi untuk menentukan lebar garis dengan panjang yang sama. Kemudian terdapat variabel x berisi np.linspace(a,b,n) yang berfungsi membuat titik sebanyak 150 antara batas a dan b dengan jarak yang sama. Kemudian terdapat sebuah perhitungan untuk mendapatkan hasil integral dari metode trapezoid, dan hasil tersebut ditampilkan menggunakan kode print.

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

Kode ini berisi variabel xp yang berfungsi membuat 1000 titik x dari a sampai b untuk membentuk sebuah kurva. Kemudian hasil tersebut dibuat kurva atau plot grafik dan ditampilkan.

```
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 = 'green')

plt.show()
```

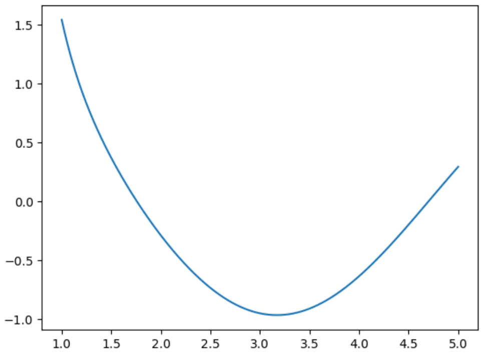
Kode ini berfungsi untuk membuat plot grafik dengan menampilkan grid yang digunakan dibawah kurva tersebut dengan warna hijau.

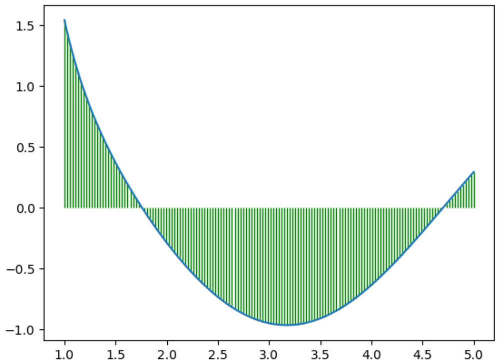
```
xp = np.linspace(a, b, 1000)
plt.plot(xp, func(xp), label="f(x)")

for i in range(1, n):
    plt.fill_between([x[i-1], x[i]], [func(x[i-1]), func(x[i])], color='green', alpha=0.4)

plt.show()
```

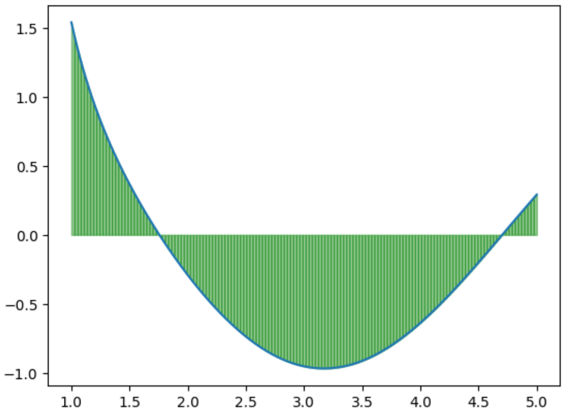
Kode ini berfungsi untuk membuat plot grafik dengan menampilkan grid yang digunakan dibawah kurva tersebut dengan warna hijau, namun bedanya terdapat visualisasi area dibawah kurva yang dihitung menggunakan metode trapezoid.

Hasil:
-1.320107290655107

Grafik:







Hasil yang didapatkan melalui metode trapezoid ini adalah -1.320107290655107 dimana metode ini dihitung menggunakan kode program, metode trapezoid ini dilakukan dengan cara membagi daerah dibawah kurva menjadi beberapa trapezoid. Kemudian luas trapezoid tersebut dihitung, dan dijumlahkan untuk mendapatkan nilai aproksimasi dari integral. Apabila dibandingkan dengan hasil yang didapat menggunakan metode eksak yaitu 1,3204 terdapat sedikit perbedaan. Perbedaan ini terjadi karena pengaruh nilai n yang digunakan pada kode program, nilai n ini adalah jumlah grid yang digunakan untuk membagi bagian bawah dari kurva, jika semakin besar nilai n maka hasil yang didapat dari metode trapezoid ini juga akan semakin akurat dan mendekati hasil dari metode eksak.

Metode Simpson

```
import numpy as np
import matplotlib.pyplot as plt
```

Kode ini berfungsi untuk mengimport library numpy dan matplotlib yang akan digunakan pada kode program. Numpy untuk perhitungan numerik, dan matplotlib untuk membuat plot grafik.

```
def func(x):
    return x**-3 + np.cos(x)

a = 1.0
b = 5.0
n = 150
```

Kode ini berfungsi untuk mendefinisikan sebuah fungsi yang akan diintegralkan, dimana fungsinya adalah $f(x) = x^{-3} + \cos(x)$ dan mendefisinikan batas atas dan batas bawah integral, dimana batas atasnya adalah 5 dan batas bawahnya adalah 1, dan juga terdapat pendefinisian jumlah garis yang akan digunakan untuk membagi area dibawah kurva, dan digunakan 150 garis.

```
if n % 2 == 0:
    n += 1

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

hasil = func(x[0]) + func(x[-1])

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

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

hasil *= dx / 3
```

```
print(hasil)
```

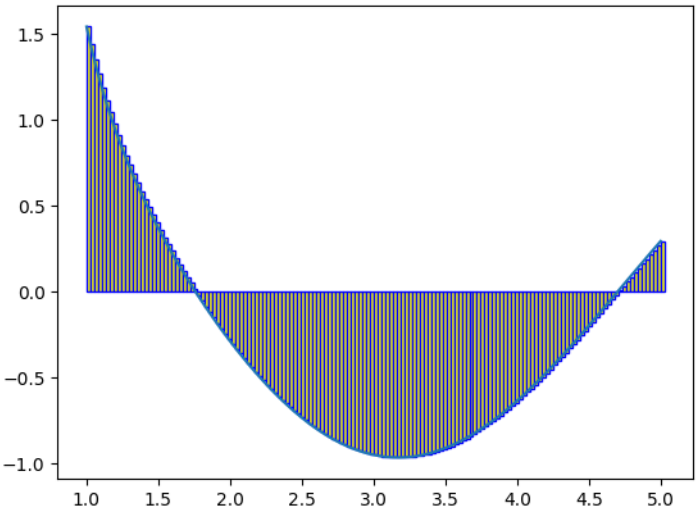
Karena metode simpson hanya dapat bekerja jika jumlah nilai n nya ganjil, maka dilakukan pemeriksaan nilai n yang digunakan dengan menggunakan logika **if n % 2 == 0** logika ini berfungsi memeriksa nilai n dengan cara membagi nilai n tersebut dengan 2 jika hasil bagi nilai n tersebut dengan 2 menghasilkan 0 maka nilai n tersebut ganjil, dan dilakukan penambahan 1 pada nilai n agar nilai n menjadi ganjil. Kemudian terdapat variabel x berisi np.linspace(a,b,n) yang berfungsi membuat titik sebanyak 151 antara batas a dan b dengan jarak yang sama. Kemudian terdapat sebuah perhitungan untuk mendapatkan hasil integral dari metode simpson, dan hasil tersebut ditampilkan menggunakan kode print.

```
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='yellow', edgecolor='blue')
plt.show()
```

Kode ini berfungsi untuk membuat plot grafik dengan menampilkan grid yang digunakan dibawah kurva tersebut dengan warna kuning dan garis kurva warna biru.

Hasil:
-1.3203950965766056

Grafik:



Hasil yang didapatkan melalui metode trapezoid ini adalah -1.3203950965766056 dimana metode ini dihitung menggunakan kode program, metode simpson ini dilakukan dengan menggunakan polynomial kuadratik untuk mendekati fungsi yang diintegralkan. Apabila dibandingkan dengan hasil yang didapat menggunakan metode eksak yaitu 1,3204 hasil ini sangat mendekati hasil dari metode eksak dengan hanya sedikit perbedaan, perbedaan tersebut terjadi karena nilai n yang digunakan, jadi semakin besar nilai n yang digunakan maka hasil yang didapatkan akan semakin akurat.

Perbedaan dari setiap metode dan mana yang lebih efektif

Perbedaan dari setiap metode yang digunakan yaitu metode eksak, metode trapezoid, dan metode simpson terletak pada cara kerja dari masing-masing metode.

Pada metode eksak digunakan aturan kalkulus dan menghitung integral secara analitik, hasil yang didapatkan dari metode eksak ini merupakan yang paling akurat dan tanpa kesalahan seperti kedua metode lainnya, namun metode ini akan sulit digunakan jika fungsi yang diberikan lebih rumit atau kompleks.

Pada metode trapezoid ini dilakukan dengan cara membagi daerah dibawah kurva menjadi beberapa trapezoid. Kemudian luas trapezoid tersebut dihitung, dan dijumlahkan untuk mendapatkan nilai aproksimasi dari integral. Metode trapezoid ini sangat sederhana untuk digunakan, namun nilai yang dihasilkan terdapat selisih yang lumayan jauh dengan hasil dari metode eksak, dan metode ini sangat bergantung dengan nilai n.

Pada metode simpson ini dilakukan dengan menggunakan polynomial kuadratik untuk mendekati fungsi yang diintegralkan. Metode simpson ini sangat kompleks dan agak sulit untuk digunakan, namun nilai yang dihasilkan menggunakan metode ini mendekati nilai yang dihasilkan melalui metode eksak.

Dari ketiga metode tersebut, metode yang paling efektif digunakan adalah metode eksak, karena hasil yang didapatkan sangat akurat dengan menggunakan aturan kalkulus, dan tidak ada selisih numerik seperti kedua metode lainnya. Namun, metode eksak ini efektif digunakan apabila fungsi tersebut sederhana, jika fungsi yang akan dihitung sangat kompleks memungkinkan waktu lebih dalam pengerjaannya, dan menurut say ajika fungsi tersebut sangat kompleks dapat digunakan metode simpson karena nilai yang dihasilkan melalui metode ini sangat mendekati hasil yang didapatkan melalui metode eksak.