

PRAKTIKUM FISIKA KOMPUTASI

INTEGRAL METODE NUMERIKAL

Disusun Oleh:

Dewi Rahmawati(1227030010)

Pada praktikum kali ini yaitu menyelesaikan persamaan integral dengan menggunakan tiga metode yaitu metode eksak metode trapezoid, dan metode simpson. Untuk penyelesaian persamaan integral menggunakan metode eksak yaitu menyelesaikannya dengan menggunakan perhitungan manual analitik dengan menggunakan aturan kalkulus, yang melibatkan penggunaan rumus integral seperti integral polinomial atau trigonometri. Perhitungan dengan metode ini menghasilkan nilai sebesar (-1,32). Perhitungannya yaitu sebagai berikut:

The image shows a handwritten solution for the exact method of integration. It starts with the title 'Metode Eksak' in pink. Below it, the integral to be solved is $\int_1^5 x^{-3} + \cos(x) dx$. The solution is written in blue ink and follows these steps: 1. Write the integral: $\int_1^5 x^{-3} + \cos(x) dx$. 2. Find the antiderivative: $\left[\frac{1}{-3+1} x^{-3+1} + \sin(x) \right]_1^5$. 3. Simplify the antiderivative: $\left[-\frac{1}{2} x^{-2} + \sin(x) \right]_1^5$. 4. Evaluate at the bounds: $\left[-\frac{1}{2x^2} + \sin(x) \right]_1^5$. 5. Calculate the values: $\left[-\frac{1}{2(5)^2} + \frac{1}{2(1)^2} + \sin(5) - \sin(1) \right]$. 6. Simplify the expression: $= -\frac{1}{50} + \frac{1}{2} + (-0,958 - 0,841)$. 7. Further simplification: $= \frac{24}{50} + (-1,8)$. 8. Final result: $= 0,48 - 1,8 = -1,32$.

Untuk metode trapezoid yaitu menggunakan pembagian area di bawah kurva menjadi trapezoid untuk menghitung aproksimasi integral. Semakin banyak trapezoid (segmen), semakin mendekati hasil integral sesungguhnya. Pada metode trapezoid ini dibantu oleh kode program yang kode program tersebut beroperasi dibantu website *colab.google*. Untuk Kode programnya yaitu

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

Kode program ini dipakai untuk mengimport library numpy dan matplotlib. Library numpy yaitu digunakan untuk membaca atau mengakses kode perhitungan numerik (rumus) yang akan dimasukkan, sedangkan *library* matplotlib berfungsi untuk membuat grafik dari hasil perhitungan yang dilakukan.

```
#Integral
```

```
def func(x):                                #Nama fungsi
    return (x**-3)+np.cos(x)                #fungsi yang akan diintegalkan
a = 1.0                                    #Batas bawah
b = 5.0                                    #Batas Atas
```

Kode program tersebut merupakan kode program untuk menginputkan persamaan yang akan diintegalkan di bagian 'def func (x):' dengan a sebagai batas bawah integral dan b sebagai batas atas integralnya.

```
#Metode Trapezoid
n = 100                                #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])

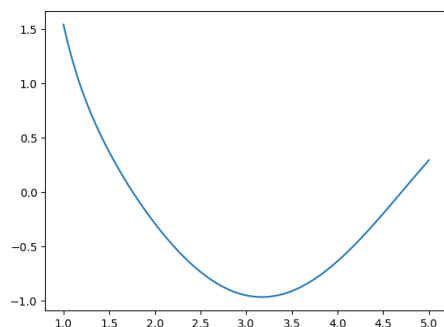
hasil = 0.5*dx*(func(x[0])+2*sigma+func(x[-1]))

print(hasil)
```

Pada bagian kode program ini terdapat persamaan dari metode trapezoid, dengan n menyatakan jumlah banyaknya grid (garis) dimana semakin banyak grid yang dimasukkan maka hasilnya akan semakin bagus dan semakin mendekati nilai yang diperoleh dari metode eksak. Untuk 'dx' yaitu menyatakan lebar dari sub interval, yakni panjang total interval dibagi dengan jumlah grid dikurangi satu. Untuk 'x' nya yaitu berupa array. Untuk 'sigma = 0' dan kebawahnya itu digunakan untuk menghitung nilai data dari setiap gridnya. Sedangkan 'print (hasil)' itu untuk memunculkan nilainya. Untuk nilai yang didapatkan pada metod ini yaitu sebesar (-1.319743079146315).

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

Kode program diatas digunakan untuk membuat grafik dari persamaan yang telah diperoleh hasilnya tadi. Grafik yang dihasilkannya yaitu



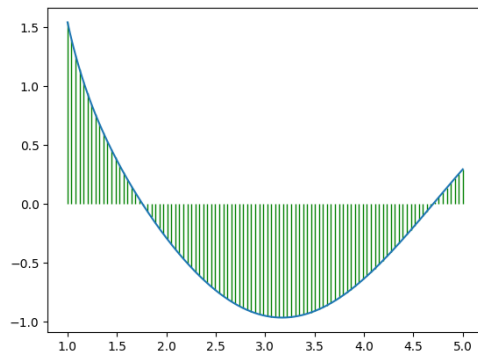
```
xp =np.linspace(a,b,1000)
```

```
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 program diatas digunakan untuk membuat grafik dengan menampilkan nilai grid nya yang menunjukkan estimasi area di bawah fungsi persamaan. Menggunakan data 1000 titik dibawah kurva. Untuk setiap gridnya diberi warna hijau. Untuk kurvanya yaitu:



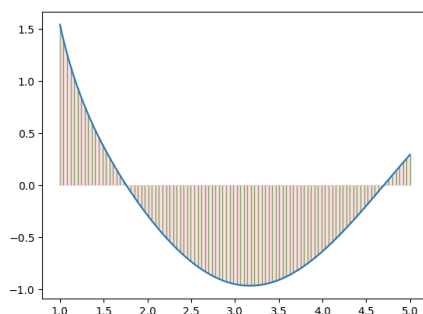
```
xp =np.linspace(a,b,1000)
plt.plot(xp, func(xp))

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

plt.fill_between(x, func(x), color= 'pink', alpha=0.5)

plt.show()
```

Kode program diatas untuk memberi arsiran daerah pada kurva yang diperoleh dari peyelesaian persamaan dengan menggunakan metode trapezoid. Untuk hasil kurvanya yaitu:



Untuk metode simpson yaitu metode yang menggunakan polinomial kuadratik (parabola) untuk mendekati fungsi yang diintegrasikan di antara segmen-segmen. Pada metode simpso juga dihitung menggunakan kode program yang prosesnya dibantu oleh website *colab.google*. Kode programnya yaitu:

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

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

#Batas Integrasi
a = 1.0
b = 5.0
n = 150
```

Kode program diatas terdiri dari peng-*import*-an *library* dan persamaan yang akan dihitungnya dengan a sebagai batas bawah, b sebagai batas atas, dan n sebagai jumlah gridnya. *Library* yang diimport yaitu *library* numpy dan matplotlib. *Library* numpy yaitu digunakan untuk membaca atau mengakses kode perhitungan numerik (rumus) yang akan dimasukkan, sedangkan *library* matplotlib berfungsi untuk membuat grafik dari hasil perhitungan yang dilakukan.

```
#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

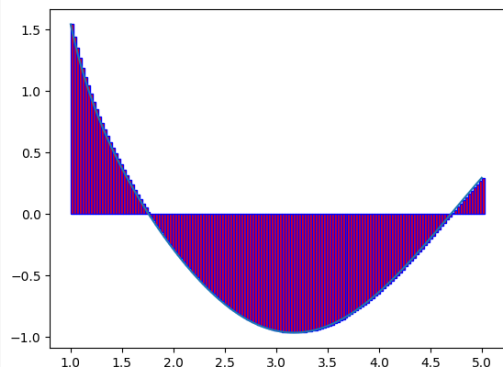
#Visualisasi grafik dan bar
xp = np.linspace(a,b,1000)
```

```
plt.plot(xp, func(xp))

for i in range(n):
    plt.bar(x[i], func(x[i]), align = 'edge', width=dx, color = 'red',
    edgecolor = 'blue')
plt.show()
print(hasil)
```

Pada kode program metode simpson nilai n nya harus ganjil, maka dilakukan pengecekan menggunakan `'if n % 2 == 0:n +=1'` menandakan jika hasil bagi dari n nya itu 0, maka n tersebut bernilai genap dan nilai n ditambah 1 supaya nilai n ganjil. Kode `"x = np.linspace(a, b, n) ; dx = (x[-1] - x[0]) / (n - 1)"` digunakan untuk menentukan n ti data dan jarak antara titik datanya. Pada bagian dibawah tagar menghitung integral menggunakan metode simpson digunakan untuk inisial nilai fungsi dan mencari indeks titik integralnya. Jika hasil titik indeksnya ganjil maka hasilnya penjumlahannya akan dikalikan dengan 4, sedangkan jika hasil titiknya genap maka hasil penjumlahannya dikalikan dengan 2. Hal tersebut sesuai dengan aturan dari metode simpson. Kemudian nilai akhirnya nanti akan dikalikan dengan 3.

Selanjutnya dibawah tagar visualisasi grafik dan bar yaitu untuk memplot hasil dari 1000 titik indeks yang dihasilkan menjadi sebuah grafik dengan grid dan arsiran. Lalu untuk baga terakhir yaitu `'print(hasil)'` untuk menampilkan hasil akhir dari perhitungan integral menggunakan metode simpsonnya. Untuk hasil perhitungan integral dengan menggunakan metode simpsonnya yaitu -1.3203950965766056 . Untuk hasil grafiknya yaitu



Pada ketiga metode tersebut dapat dilihat bahwa hasil penyelesaian persamaan dengan menggunakan metode eksak yaitu $-(1,32)$. Untuk penyelesaian menggunakan metode trapezoid dengan bantuan colab.google yaitu didapat hasilnya sebesar (-1.319743079146315) . Sedangkan untuk metode simpson hasil penyelesaian persamaannya adalah -1.3203950965766056 . Dapat dilihat bahwa untuk metode trapezoid dan metode simpson mendekati hasil yang diperoleh dengan menggunakan metode eksak. Perbedaan nilai antara metode trapezoid dan metode simpson dipengaruhi oleh nilai grid (n) yang di masukkan. Semakin besar nilai grid yang dimasukkan maka hasilnya akan semakin mendekati metode eksak. Untuk hasil grafik pada metode trapezoid dan simpson sama yaitu mempunyai kurva landay kebawah dibawah nol. Hal tersebut dikarenakan hasil yang diperoleh dari persamaan tersebut menunjukkan hasil yang minus.

Perbedaan antara metode eksak, trapezoid, dan simpson yaitu ada pada cara perhitungannya masing-masing. Untuk metode eksak cara perhitungannya yaitu menyelesaikan dengan menggunakan

perhitungan manual analitik dengan menggunakan aturan kalkulus, yang melibatkan penggunaan rumus integral seperti integral polinomial atau trigonometri. Untuk metode trapezoid yaitu perhitungannya menggunakan pembagian area di bawah kurva menjadi trapezoid untuk menghitung aproksimasi integral. Semakin banyak trapezoid (segmen), semakin mendekati hasil integral sesungguhnya. Sedangkan untuk metode simpson yaitu metode yang menggunakan polinomial kuadrat (parabola) untuk mendekati fungsi yang diintegralkan di antara segmen-segmennya. Dari ketiga metode tersebut mempunyai kelebihan dan kekurangannya sendiri. Untuk perhitungan yang paling efektif menurut saya ketika menguasai semua rumus dan perhitungan dalam kalkulusnya maka metode eksaklah yang paling efektif karena hasil yang dihasilkan adalah hasil yang tepat bukan merupakan hasil aproksiasi. Akan tetapi metode eksak rawan menghasilkan hasil yang salah ketika orang yang menghitung solusi persamaan tersebut kurang mampu menguasai rumus dan cara perhitungannya. Terlebih jika orang yang menghitungnya adalah orang yang ceroboh. Maka dari itu untuk penyelesaian persamaan bagi orang yang kurang menguasai kalkulus menurut saya metode simpson lebih bagus karena mampu menyelesaikan persamaan yang kompleks dengan lebih cepat.